

# The Zig-Zag Process and Super-Efficient Sampling for Bayesian Analysis of Big Data

Joris Bierkens<sup>\*1</sup>, Paul Fearnhead<sup>\*2</sup> and Gareth Roberts<sup>\*1</sup>

<sup>1</sup>*Department of Statistics,  
University of Warwick,  
Coventry CV4 7AL,  
United Kingdom*

<sup>2</sup>*Department of Mathematics and Statistics,  
Fylde College,  
Lancaster University,  
Lancaster, LA1 4YF,  
United Kingdom*

**Abstract:** Standard MCMC methods can scale poorly to big data settings due to the need to evaluate the likelihood at each iteration. There have been a number of approximate MCMC algorithms that use sub-sampling ideas to reduce this computational burden, but with the drawback that these algorithms no longer target the true posterior distribution. We introduce a new family of Monte Carlo methods based upon a multi-dimensional version of the Zig-Zag process of [Bierkens and Roberts \(2016\)](#), a continuous time piecewise deterministic Markov process. While traditional MCMC methods are reversible by construction (a property which is known to inhibit rapid convergence) the Zig-Zag process offers a flexible non-reversible alternative which we observe to often have favourable convergence properties. The dynamics of the Zig-Zag process correspond to a constant velocity model, with the velocity of the process switching at events from a point process. The rate of this point process can be related to the invariant distribution of the process. If we wish to target a given posterior distribution, then rates need to be set equal to the gradient of the log of the posterior. Unlike traditional MCMC, We show how the Zig-Zag process can be simulated without discretisation error, and give conditions for the process to be ergodic. Most importantly, we introduce a sub-sampling version of the Zig-Zag process that is an example of an *exact approximate scheme*. That is, if we replace the true gradient of the log posterior with an unbiased estimator, obtained by sub-sampling, then the resulting approximate process still has the posterior as its stationary distribution. Furthermore, if we use a control-variate idea to reduce the variance of our unbiased estimator, then both heuristic arguments and empirical observations show that Zig-Zag can be super-efficient: after an initial pre-processing step, essentially independent samples from the posterior distribution are obtained at a computational cost which does not depend on the size of the data.

Primary 65C60; secondary 65C05, 62F15, 60J25.

**Keywords and phrases:** MCMC, non-reversible Markov process, piecewise deterministic Markov process, Stochastic Gradient Langevin Dynamics, sub-sampling, exact sampling.

---

<sup>\*</sup>The authors acknowledge the EPSRC for support under grants EP/D002060/1 (CRiSM) and EP/K014463/1 (iLike)

## 1. Introduction

The importance of Markov chain Monte Carlo techniques in Bayesian inference shows no signs of diminishing. However, despite an industry of elaborations, all commonly used methods are variants on the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) and rely on innovations which date back over 60 years. All MH algorithms essentially simulate realisations from a discrete reversible ergodic Markov chain with invariant distribution  $\pi$  which is (or is closely related to) the *target* distribution, i.e. the posterior distribution in a Bayesian context. The MH algorithm gives a beautifully simply though flexible recipe for constructing Markov chains with the right invariant properties, requiring only local information about  $\pi$  (typically pointwise evaluations of  $\pi$  and, in certain implementations of the algorithm, its derivative at the current and proposed new locations) to complete each iteration.

However new complex modelling and data paradigms are seriously challenging these established methodologies. Firstly, the restriction of traditional MCMC to reversible Markov chains is a serious limitation. It is now well-understood both theoretically (Hwang, Hwang-Ma and Sheu, 1993; Sun, Gomez and Schmidhuber, 2010; Chen and Hwang, 2013; Rey-Bellet and Spiliopoulos, 2015; Bierkens, 2015; Lelièvre, Nier and Pavliotis, 2013; Duncan, Lelièvre and Pavliotis, 2016) and heuristically (Neal, 1998) that non-reversible chains offer potentially massive advantages over reversible counterparts. The need to escape reversibility, and create momentum to aid mixing throughout the state space is certainly well-known, and motivates a number of the most ingenious modern MCMC methods, including the popular Hamiltonian MCMC (HMC, Duane et al. (1987); Neal (2011)). Inspired by analogy to Hamiltonian dynamics, HMC works in discrete time by approximating the trajectories of Hamiltonian flow, and using these as proposals within a MH algorithm. Whilst the proposals are based on the non-reversible Hamiltonian dynamics, the resulting MH algorithm turns out to be reversible, albeit on an enlarged state space.

Until a recent breakthrough (Turitsyn, Chertkov and Vucelja, 2011) it has not been possible to construct generic non-reversible MCMC methods. Turitsyn, Chertkov and Vucelja (2011) introduce a general framework for *lifted* Markov chains which embeds the distribution of interest in a space of higher dimension, incorporating in addition a velocity component designed to create momentum through the state space and break down *random-walk*-type behaviour of the chain. In this way it brings to fruition the ideas first postulated and studied in simple cases in Diaconis, Holmes and Neal (2000). A remaining difficulty of the algorithm of Turitsyn, Chertkov and Vucelja (2011) is that the method depends on choosing a quantity which determines the type of momentum generated, the selection of which is non-trivial and of significant influence on the algorithmic efficiency.

In Bierkens and Roberts (2016), the application of Turitsyn, Chertkov and Vucelja (2011) to a popular model in statistical physics, the Curie-Weiss model, was analysed, and its high-dimensional limit was shown to behave like a continuous-time piecewise deterministic stochastic process termed the *Zig-Zag process*. We

shall see that the Zig-Zag process provides a practically implementable algorithm with some remarkable properties.

A second major obstacle to the application of MCMC for Bayesian inference in challenging problems is the need to process potentially massive data-sets. It can be impractical to carry out large numbers of MH iterations in reasonable time scales. This has led to a range of alternatives to MH that use sub-samples of the data at each iteration (Welling and Teh, 2011; Ma, Chen and Fox, 2015; Quiroz, Villani and Kohn, 2015), or that partition the data into shards, run MCMC on each shard, and then attempt to combine the information from these different MCMC runs (Neiswanger, Wang and Xing, 2013; Scott, Blocker and Bonassi, 2016; Minsker et al., 2014; Wang and Dunson, 2013; Srivastava et al., 2015). However all of these methods introduce some form of approximation error. The final sample will be drawn from some approximation to the posterior, and the quality of the approximation can be impossible to evaluate.

This paper introduces the multi-dimensional Zig-Zag sampling algorithm (ZZ) and its variants (ZZ-SS, ZZ-CV). These methods overcome the restrictions of the lifted Markov chain approach of Turitsyn, Chertkov and Vucelja (2011) as they do not depend on the introduction of momentum generating quantities. It is also amenable to the use of sub-sampling ideas. The dynamics of the Zig Zag process depends on the target distribution through the gradient of the logarithm of the target. For Bayesian applications this is a sum, and is easy to estimate unbiasedly using subsampling. Moreover, Zig-Zag with Sub-Sampling (ZZ-SS) retains the exactness of the required invariant distribution. Furthermore, if we also use control variate ideas, to reduce the variance of our subsampling estimator of the gradient, the resulting Zig Zag with Control Variates (ZZ-CV) algorithm has remarkable *super-efficient* scaling properties for large data sets.

We will call an algorithm *super-efficient* if it is able to generate independent samples from the target distribution at a higher efficiency than if we would draw from the target distribution at the cost of evaluating all data. The only situation we are aware of where we can implement super-efficient sampling is with simple conjugate models, where the likelihood function has a low-dimensional summary statistic. In this case, the cost of computing the parameters of the posterior distribution is  $O(n)$ , where  $n$  is the number of observations. Once we have performed this pre-computation, we can obtain independent samples from the posterior distribution at a cost of  $O(1)$ , by using the functional form of the posterior distribution with the pre-computed parameters inserted. In applied statistical settings it is usually not feasible to work with conjugate prior distributions. In these situations, standard Monte Carlo methods require us to evaluate all observations at every iteration, and each iteration will be of  $O(n)$ . By comparison, ZZ-CV can be super-efficient, in that it replicates the computational efficiency of working with a conjugate prior distribution: after a pre-computation of  $O(n)$ , we are able to obtain independent samples at a cost of  $O(1)$ .

This breakthrough is based upon the Zig-Zag process, a continuous time piecewise deterministic Markov process (PDMP), with trajectories which we will now briefly describe. Given a  $d$ -dimensional target density  $\pi$ , assumed to be

differentiable and positive, Zig-Zag is a continuous-time non-reversible stochastic process with continuous and piecewise linear trajectories on  $\mathbb{R}^d$ . It moves with constant velocity,  $\Theta \in \{-1, 1\}^d$ , until a change of direction event occurs at which one of the velocity components switches sign. The event time and choice of which direction to reverse is controlled by a collection of state-dependent switching rates,  $(\lambda_i)_{i=1}^d$  which in turn are constrained via an identity (3) which ensures that  $\pi$  is a stationary distribution for the process. The process intrinsically is constructed in continuous-time, and it can be easily simulated using standard Poisson thinning arguments as we shall see in Section 3.

The use of piecewise deterministic Markov processes (PDMPs) such as the Zig-Zag processes is an exciting and mostly unexplored area in MCMC. The first occurrence of a PDMP for sampling purposes is in the computational physics literature (Peters and De With (2012)), which in one dimension coincides with the Zig-Zag process. In Bouchard-Côté, Vollmer and Doucet (2015) this method is given the name *Bouncy Particle Sampler*, analysed in some detail and extended in several directions. In multiple dimensions the Zig-Zag process and Bouncy Particle Sampler (BPS) are different processes: both are PDMPs which move along straight line segments, but the Zig-Zag process changes direction in only a single component at each switch, whereas the Bouncy Particle Sampler reflects the full direction vector in the level curves of the density function. As we will see in Section 2.4, this difference seems to have a beneficial effect on the ergodic properties of the Zig-Zag process. The one-dimensional Zig-Zag process is analysed in detail in e.g. Fontbona, Guérin and Malrieu (2012); Monmarché (2014); Fontbona, Guérin and Malrieu (2016); Bierkens and Roberts (2016). Historically Goldstein (1951); Kac (1974) introduced the Zig-Zag process with constant switching rates, known since then as the telegraph process.

A continuous-time sequential Monte Carlo algorithm for scalable Bayesian inference with big data (the SCALE algorithm) is given in Pollock et al. (2016), based on the dynamic propagation of weights on a simulated diffusion sample path. Although both SCALE and Zig-Zag have similar motivation and are intrinsically continuous-time in their approaches, they are otherwise very different, and both methods have their clear advantages. One important advantage that Zig-Zag has over SCALE is that it avoids the issue of controlling the stability of importance weights. It also has the advantage of being simpler to implement. On the other hand the SCALE algorithm has the property that it is well-adapted for the use of parallel architecture computing, and has particularly simple scaling properties for big data.

We shall structure the paper as follows. In Section 2 we shall introduce the canonical Zig-Zag property and explore some of its basic properties. Section 3 describes various strategies for implementing the Zig-Zag in practice, all based around Poisson thinning ideas. The Zig-Zag is then extended in Section 4 to the context where the the gradient  $\nabla \log \pi$  is intractable but can be readily estimated unbiasedly. This is applied to the big data context via sub-sampling and an order of magnitude efficiency gain is subsequently achieved by a further control variate modification. In Section 5 we will describe the behaviour of the computing costs of implementing the algorithm (incorporating both algorithm

convergence time and computation costs) scales with the size of the data set for the ZZ and ZZ-SS algorithms. These results are supported through experiments and examples in Section 6 including a favourable comparison with the recently popular Stochastic Gradient Langevin Dynamics approximation method for big data MCMC (Welling and Teh (2011); Teh, Thiery and Vollmer (2014)).

### 1.1. Frequently used notation

For a topological space  $X$  let  $C(X)$  denote the space of continuous functions on  $X$  and let  $\mathcal{B}(X)$  denote the Borel  $\sigma$ -algebra. We write  $\mathbb{R}_+ := [0, \infty)$ . If  $h : \mathbb{R}^d \rightarrow \mathbb{R}$  is differentiable then  $\partial_i h$  denotes the function  $\xi \mapsto \frac{\partial h(\xi)}{\partial \xi_i}$ . Throughout this paper we will work with the topological space  $E = \mathbb{R}^d \times \{-1, +1\}^d$ , where the topology is the product topology of the Euclidean topology on  $\mathbb{R}^d$  and the discrete topology on  $\{-1, +1\}^d$ . Elements in  $E$  will often be denoted by  $(\xi, \theta)$  with  $\xi \in \mathbb{R}^d$  and  $\theta \in \{-1, +1\}^d$ . For  $g : E \rightarrow \mathbb{R}$  differentiable in its first argument we will use the shorthand notation  $\partial_i g$  to denote the function  $(\xi, \theta) \mapsto \frac{\partial g(\xi, \theta)}{\partial \xi_i}$ ,  $i = 1, \dots, d$ .

## 2. The Zig-Zag process

We will first define the Zig-Zag process via its generator, before giving an informal description of the process. We will then describe the dynamics of the process more formally through a general recipe for simulating this continuous-time stochastic process.

For  $k \in \{1, \dots, d\}$ , let  $F_k : \{-1, +1\}^d \rightarrow \{-1, +1\}^d$  denote the operation of flipping the  $k$ -th bit in a binary vector  $\theta \in \{-1, +1\}^d$ , i.e.

$$(F_k[\theta])_i := \begin{cases} \theta_i & i \neq k \\ -\theta_i & i = k. \end{cases}$$

Let  $\lambda \in C(E; \mathbb{R}_+^d)$ ; we will refer to  $\lambda$  as the *switching rate* throughout this paper. Define a densely defined operator  $L$  on  $C(E)$  by

$$Lf(\xi, \theta) = \sum_{i=1}^d \{\theta_i \partial_i f(\xi, \theta) + \lambda_i(\xi, \theta)(f(\xi, F_i[\theta]) - f(\xi, \theta))\}, \quad (\xi, \theta) \in E, \quad (1)$$

for  $f \in C(E)$  such that  $f(\cdot, \theta)$  has compact support and is differentiable for all  $\theta \in \{-1, +1\}^d$ .

The operator  $L$ , extended to its maximal domain  $\mathcal{D}(L)$ , is the generator of a piecewise deterministic Markov process satisfying the strong Markov property (Davis (1984)). The trajectories will be denoted by  $(\Xi(t), \Theta(t))_{t \geq 0}$  and can be described as follows: at random times a single component of  $\Theta(t)$  flips. In between these switches,  $\Xi(t)$  is linear with  $\frac{d}{dt}\Xi(t) = \Theta(t)$ . The rates at which the flips in  $\Theta(t)$  occur are time inhomogeneous: the  $i$ -th component of  $\Theta$  switches at rate  $\lambda_i(\Xi(t), \Theta(t))$ .

### 2.1. Construction

For a given  $(\xi, \theta) \in E$ , we may construct a trajectory of  $(\Xi, \Theta)$  of the Markov process with generator  $L$  and initial condition  $(\xi, \theta)$  as follows.

- Let  $(T^0, \Xi^0, \Theta^0) := (0, \xi, \theta)$ .
- For  $k = 1, 2, \dots$ 
  - Let  $\xi^k(t) := \Xi^{k-1} + \Theta^{k-1}t$ ,  $t \geq 0$
  - For  $i = 1, \dots, d$ , let  $\tau_i^k$  be distributed according to

$$\mathbb{P}(\tau_i^k \geq t) = \exp\left(-\int_0^t \lambda_i(\xi^k(s), \Theta^{k-1}) ds\right).$$

- Let  $i_0 := \operatorname{argmin}_{i \in \{1, \dots, d\}} \tau_i^k$  and let  $T^k := T^{k-1} + \tau_{i_0}^k$ .
- Let  $\Xi^k := \xi^k(T^k)$ .
- Let

$$\Theta^k(i) := \begin{cases} \Theta^{k-1}(i) & \text{if } i \neq i_0, \\ -\Theta^{k-1}(i) & \text{if } i = i_0 \end{cases}$$

This procedure defines a sequence of *skeleton points*  $(T^k, \Xi^k, \Theta^k)_{k=0}^\infty$  in  $\mathbb{R}_+ \times E$ , which correspond to the time and position at which the direction of the process changes. The trajectory  $\xi^k(t)$  represents the position of the process at time  $T^{k-1} + t$  until time  $T^k$  (ie for  $0 \leq t \leq T^k - T^{k-1}$ ). The time until the next skeleton event is characterized as the smallest time of a set of events in  $d$  simultaneous point processes, where each point process corresponds to switching events of a different component of the velocity. For the  $i$ -th of these point processes, events occur at rate  $\lambda_i(\xi^k(s), \Theta^{k-1})$ , and  $\tau_i^k$  is defined to be the time to the first event for the  $i$ -th component. The component for which the earliest event occurs is indicated by  $i_0$ . This both defines  $\tau_{i_0}^k$ , the time between the  $(k-1)$ th and  $k$ th skeleton point, and the component  $i_0$  of the velocity that switches.

The piecewise deterministic trajectories  $(\Xi(t), \Theta(t))$  are now obtained as

$$(\Xi(t), \Theta(t)) := (\Xi^k + \Theta^k(t - T^k), \Theta^k) \quad \text{for } t \in [T^k, T^{k+1}), \quad k = 0, 1, 2, \dots$$

Since the switching rates are continuous and hence bounded on compact sets, and since  $\Xi$  will travel a fixed finite distance within any finite time interval, within any bounded time interval there will be only finitely many switches almost surely.

The above procedure provides a mathematical construction of a Markov process with  $L$  as its generator, as well as (almost) an algorithm which simulates this process. The only step in this procedure which presents a computational challenge is the simulation of the random times  $(T_i^k)$  and a significant part of this paper will consider obtaining these in a numerically efficient way.

In Figure 1 trajectories of the Zig-Zag process are displayed for a few examples of invariant distributions. The name of the process is derived by the *zig-zag* nature of paths that the process produces.

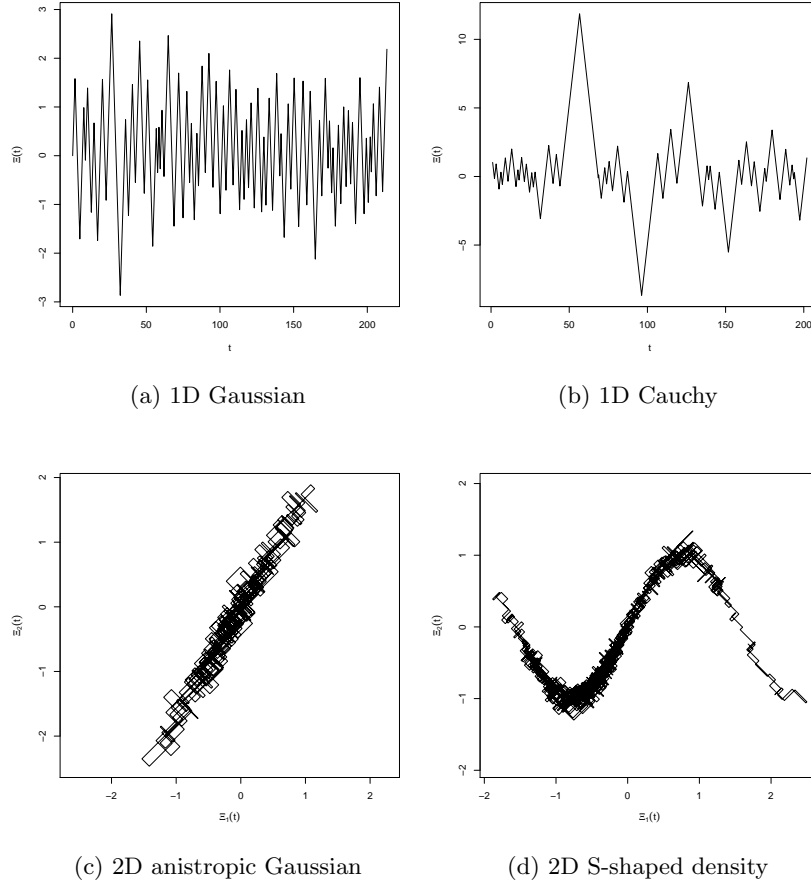


Figure 1: Example trajectories of the canonical Zig-Zag process. In the one-dimensional examples (a) and (b), the horizontal axis shows time and the vertical axis the  $\Xi$ -coordinate of the process. In the two-dimensional examples (c) and (d), the trajectories in  $\mathbb{R}^2$  of  $(\Xi_1, \Xi_2)$  are plotted.

## 2.2. Invariant distribution

The most important aspect of the Zig-Zag process is that in many cases the switching rates are directly related to an easily identifiable invariant distribution. Let  $C^1(\mathbb{R}^d)$  denote the space of continuously differentiable functions on  $\mathbb{R}^d$ . We introduce the following assumption.

**Assumption 2.1.** *For some function  $\Psi \in C^1(\mathbb{R}^d)$  satisfying*

$$\int_{\mathbb{R}^d} \exp(-\Psi(\xi)) \, d\xi < \infty \quad (2)$$

*we have*

$$\lambda_i(\xi, \theta) - \lambda_i(\xi, F_i[\theta]) = \theta_i \partial_i \Psi(\xi) \quad \text{for all } (\xi, \theta) \in E, i = 1, \dots, d. \quad (3)$$

Throughout this paper we will often refer to  $\Psi$  as the *negative log density*. Let  $\mu_0$  denote the measure on  $\mathcal{B}(E)$  such that, for  $A \in \mathcal{B}(\mathbb{R}^d)$  and  $\theta \in \{-1, +1\}^d$ ,

$$\mu_0(A \times \{\theta\}) = \text{Leb}(A),$$

with  $\text{Leb}$  denoting Lebesgue measure on  $\mathbb{R}^d$ .

**Theorem 2.2.** *Suppose Assumption 2.1 holds. Let  $\mu$  denote the probability distribution on  $E$  such that  $\mu$  has Radon-Nikodym derivative given by*

$$\frac{d\mu}{d\mu_0}(\xi, \theta) = \frac{\exp(-\Psi(\xi))}{Z}, \quad (\xi, \theta) \in E, \quad (4)$$

*where  $Z = \int_E \exp(-\Psi(\xi)) \, \mu_0(d\xi \otimes d\theta)$ . Then the Markov process  $(\Xi, \Theta)$  with generator  $L$  has invariant distribution  $\mu$ .*

*Proof.* Write  $L_i f(\xi, \theta) = \theta_i \partial_i f(\xi, \theta) + \lambda_i(\xi, \theta)(f(\xi, F_i[\theta]) - f(\xi, \theta))$ , so that  $L = L_1 + \dots + L_d$ . Let  $f \in \mathcal{D}(L)$ . Then for  $i = 1, \dots, d$ ,

$$\begin{aligned} & \int_E L_i f(\xi, \theta) \, d\mu \\ &= \frac{1}{Z} \sum_{\theta \in \{-1, +1\}^d} \int_{\mathbb{R}^d} \{\theta_i \partial_i f(\xi, \theta) + \lambda_i(\xi, \theta)(f(\xi, F_i[\theta]) - f(\xi, \theta))\} \exp(-\Psi(\xi)) \, d\xi \\ &= \frac{1}{Z} \sum_{\theta \in \{-1, +1\}^d} \int_{\mathbb{R}^d} \{-\theta_i \partial_i \Psi(\xi) + \lambda_i(\xi, F_i[\theta]) - \lambda_i(\xi, \theta)\} f(\xi, \theta) \exp(-\Psi(\xi)) \, d\xi \\ &= 0. \end{aligned}$$

Hence  $\int_E Lf \, d\mu = 0$ , which by (Ethier and Kurtz, 2005, Theorem 4.9.17) establishes invariance of  $\mu$ .  $\square$



We see that under the invariant distribution of the Zig-Zag process,  $\xi$  and  $\theta$  are independent of each other, with  $\xi$  having density proportional to  $\exp(-\Psi(\xi))$  and  $\theta$  having a uniform distribution on the points in  $\{-1, +1\}^d$ .

For  $a \in \mathbb{R}$ , let  $(a)^+ := \max(0, a)$  and  $(a)^- := \max(0, -a)$  denote the positive and negative parts of  $a$ , respectively. We will often use the trivial identity  $a = (a)^+ - (a)^-$  without comment. The following result characterizes the switching rates for which (3) holds.

**Proposition 2.3.** *Suppose  $\lambda : E \rightarrow \mathbb{R}_+^d$  is continuous. Then Assumption 2.1 is satisfied if and only if there exists a continuous function  $\gamma : E \rightarrow \mathbb{R}_+^d$  such that for all  $i = 1, \dots, d$  and  $(\xi, \theta) \in E$ ,  $\gamma_i(\xi, \theta) = \gamma_i(\xi, F_i[\theta])$  and, for  $\Psi \in C^1(\mathbb{R}^d)$  satisfying (2),*

$$\lambda_i(\xi, \theta) = (\theta_i \partial_i \Psi(\xi))^+ + \gamma_i(\xi, \theta). \quad (5)$$

*Proof.* It is straightforward to verify that if  $\lambda$  satisfies (5), with  $\gamma$  as specified, then it also satisfies (3). Conversely, suppose  $\lambda$  satisfies (3) and define

$$\gamma_i(\xi, \theta) := \lambda_i(\xi, \theta) - (\theta_i \partial_i \Psi(\xi))^+, \quad i = 1, \dots, n, (\xi, \theta) \in E.$$

Then a straightforward computation yields  $\gamma_i(\xi, \theta) - \gamma_i(\xi, F_i[\theta]) = 0$ . Now suppose for some  $(\xi, \theta) \in E$ , and  $i = 1, \dots, n$ ,  $\gamma_i(\xi, \theta) < 0$ . First suppose  $\theta_i \partial_i \Psi(\xi) \leq 0$ . Then  $\lambda_i(\xi, \theta) = \gamma_i(\xi, \theta) + 0 < 0$  which is in contradiction with the requirement that  $\lambda_i(\xi, \theta) \geq 0$ . On the other hand, if  $\theta_i \partial_i \Psi(\xi) > 0$ , then  $\lambda_i(\xi, F_i[\theta]) = \gamma_i(\xi, F_i[\theta]) + 0 = \gamma_i(\xi, \theta) < 0$ , again a contradiction. It follows that  $\gamma_i(\xi, \theta) \geq 0$  for all  $i = 1, \dots, n$ ,  $(\xi, \theta) \in E$ .  $\square$

*Remark 2.4.* The definition of the Zig-Zag process can be extended to have different speed in different directions, i.e. with a generator of the form

$$Lf(\xi, \theta) = \sum_{i=1}^d \{ \theta_i a_i \partial_i f(\xi, \theta) + \lambda_i(\xi, \theta) (f(\xi, F_i[\theta]) - f(\xi, \theta)) \}, \quad (\xi, \theta) \in E, \varphi \in \mathcal{D}(L),$$

where  $a_i > 0$  for  $i = 1, \dots, d$ . In this case  $\mu$  as in Theorem 2.2 is invariant if and only if

$$\lambda_i(\xi, \theta) - \lambda_i(\xi, F_i[\theta]) = a_i \partial_i \Psi(\xi).$$

Note that after a rescaling the Zig-Zag process with generator (1) is obtained. We will not consider this additional flexibility in this paper to keep the exposition as simple as possible.

### 2.3. Zig-Zag process for Bayesian inference

One application of the Zig-Zag process is as an alternative to MCMC for sampling from posterior distributions in Bayesian Statistics. We show here that it is straightforward to derive a class of Zig-Zag processes that have a given posterior distribution as their invariant distribution. Importantly, the dynamics of the Zig-Zag process only depend on knowing the posterior distribution up to a constant of proportionality.

To keep notation consistent with that used for the Zig-Zag process, let  $\xi \in \mathbb{R}^d$  denote a vector of continuous parameters. We are given a prior density function for  $\xi$ , which we denote by  $\pi_0(\xi)$ , and observations  $x^{1:n} = (x^1, \dots, x^n)$ . Our model for the data defines a likelihood function  $L(x^{1:n}|\xi)$ . Thus the posterior density function is

$$\pi(\xi) \propto \pi_0(\xi)L(x^{1:n}|\xi).$$

We can write  $\pi(\xi)$  in the form of the previous section,

$$\pi(\xi) = \frac{1}{Z} \exp(-\Psi(\xi)), \quad \xi \in \mathbb{R}^d,$$

where  $\Psi(\xi) = -\log \pi_0(\xi) - \log L(x^{1:n}|\xi)$ , and  $Z = \int_{\mathbb{R}^d} \exp(-\Psi(\xi)) d\xi$  is the unknown normalising constant. Now assuming that  $\log \pi_0(\xi)$  and  $\log L(x^{1:n}|\xi)$  are both continuously differentiable with respect to  $\xi$ , we have from (5) that a Zig-Zag process with rates

$$\lambda_i(\xi, \theta) = (\theta_i \partial_i \Psi(\xi))^+$$

will have the posterior density  $\pi(\xi)$  as the marginal of its invariant distribution  $\mu$ . We call the Zig-Zag process with these rates the *Canonical Zig-Zag process* for the negative log density  $\Psi$ . As explained in Proposition 2.3, we can construct a family of Zig-Zag processes with the same invariant distribution by choosing any set of functions  $\gamma_i(\xi, \theta)$ , for  $i = 1, \dots, d$ , which take non-negative values and for which  $\gamma_i(\xi, \theta) = \gamma_i(\xi, F_i[\theta])$ , and setting

$$\lambda_i(\xi, \theta) = (\theta_i \partial_i \Psi(\xi))^+ + \gamma_i(\xi, \theta), \text{ for } i = 1, \dots, d.$$

The intuition here is that  $\lambda_i(\xi, \theta)$  is the rate at which we transition from  $\theta$  to  $F_i[\theta]$ . The condition  $\gamma_i(\xi, \theta) = \gamma_i(\xi, F_i[\theta])$  means that we increase by the same amount both the rate at which we will transition from  $\theta$  to  $F_i[\theta]$  and vice versa. As our invariant distribution places the same probability of being in a state with velocity  $\theta$  as that of being in state  $F_i[\theta]$ , these two changes in rate cancel out in terms of their effect on the invariant distribution. However, changing the rates in this way does impact the dynamics of the process, with larger  $\gamma_i$  values corresponding to more frequent changes in the velocity,  $\theta$ , of the Zig-Zag process. Thus intuitively we would expect the resulting process to mix more slowly than the canonical Zig-Zag process.

Under the assumption that the Zig-Zag process has the desired invariant distribution and that it is ergodic, it follows from the Birkhoff ergodic theorem that for any bounded continuous function  $f : E \rightarrow \mathbb{R}$ ,

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t f(\Xi(s), \Theta(s)) ds = \int_E f d\mu,$$

for any initial condition  $(\xi, \theta) \in E$ . Sufficient conditions for ergodicity will be discussed in the following section. Let us mention that taking  $\gamma$  to be positive and bounded everywhere, ensures ergodicity of the Zig-Zag process, as will be established in Theorem 2.11.

## 2.4. Ergodicity of the Zig-Zag process

We have established in Section 2.2 that for any continuously differentiable, positive density  $\pi$  on  $\mathbb{R}^d$  a Zig-Zag process can be constructed that has  $\pi$  as its marginal stationary density with respect to the spatial coordinate  $\xi$ . In order for ergodic averages  $\frac{1}{T} \int_0^T f(\Xi(s)) ds$  of the Zig-Zag process to converge asymptotically to  $\pi(f)$ , we further require  $(\Xi(t), \Theta(t))$  to be ergodic, i.e. to admit a *unique* invariant distribution. This section addresses ergodicity of the Zig-Zag process and is independent of other sections so can be skipped if desired.

The issue of ergodicity is directly related to the requirement that  $(\Xi(t), \Theta(t))$  is irreducible, i.e. the state space is not reducible into components which are each invariant for the process  $(\Xi(t), \Theta(t))$ . For the one-dimensional Zig-Zag process, (exponential) ergodicity has already been established under mild conditions (Bierkens and Roberts (2016)). As we will discuss below, irreducibility and thus ergodicity can be established for large classes of multi-dimensional target distributions, such as i.i.d. Gaussian distributions, and also if the switching rates  $\lambda_i(\xi, \theta)$  are positive for all  $i = 1, \dots, d$ , and  $(\xi, \theta) \in E$ .

Let  $P^t((\xi, \theta), \cdot)$  denote the transition kernel of the Zig-Zag process with initial condition  $(\xi, \theta)$ , i.e.

$$P^t((\xi, \theta), A) = \mathbb{P}((\Xi(t), \Theta(t)) \in A \mid \Xi(0) = \xi, \Theta(0) = \theta), \quad A \in \mathcal{B}(E).$$

A function  $f : E \rightarrow \mathbb{R}$  is called *norm-like* if  $\lim_{\|\xi\| \rightarrow \infty} f(\xi, \theta) = \infty$  for all  $\theta \in \{-1, +1\}^d$ . Let  $\|\cdot\|_{\text{TV}}$  denote the total variation norm on the space of signed measures. First we consider the one-dimensional case.

**Assumption 2.5.** Suppose  $d = 1$  and there is a constant  $\xi_0 > 0$  such that

- (i)  $\inf_{\xi \geq \xi_0} \lambda(\xi, +1) > \sup_{\xi \geq \xi_0} \lambda(\xi, -1)$ , and
- (ii)  $\inf_{\xi \leq -\xi_0} \lambda(\xi, -1) > \sup_{\xi \leq -\xi_0} \lambda(\xi, +1)$ .

**Proposition 2.6.** (Bierkens and Roberts, 2016, Theorem 5) Suppose Assumption 2.5 holds. Then there exists a function  $f : E \rightarrow [1, \infty)$  which is norm-like such that the Zig-Zag process is  $f$ -exponentially ergodic, i.e. there exists a constant  $\kappa > 0$  and  $0 < \rho < 1$  such that

$$\|P^t((\xi, \theta), \cdot) - \pi\|_{\text{TV}} \leq \kappa f(\xi, \theta) \rho^t \quad \text{for all } (\xi, \theta) \in E \text{ and } t \geq 0.$$

*Example 2.7.* As an example of fundamental importance, which will also be used in the proof of Theorem 2.11, consider a one-dimensional Gaussian distribution. For simplicity let  $\pi(\xi)$  be centred,  $\pi(\xi) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\xi^2}{2\sigma^2}\right)$  for some  $\sigma > 0$ . According to (5) the switching rates take the form

$$\lambda(\xi, \theta) = (\theta\xi/\sigma^2)^+ + \gamma(\xi), \quad (\xi, \theta) \in E.$$

As long as  $\gamma$  is bounded from above, Assumption 2.5 is satisfied. In particular this holds if  $\gamma$  is equal to a non-negative constant.

*Remark 2.8.* We say a probability density function  $\pi$  is of *product form* if  $\pi(\xi) = \prod_{i=1}^d \pi_i(\xi_i)$ , where  $\pi_i : \mathbb{R}^d \rightarrow (0, \infty)$  are one-dimensional probability density functions. One of the key properties of the Zig-Zag process is that when its target density is of product form it can be seen as a product of independent Zig-Zag processes. In this case the negative log density is of the form  $\Psi(\xi) = \sum_{i=1}^d \Psi_i(\xi_i)$  and hence the switching rate for the  $i$ -th component of  $\theta$  is given by

$$\lambda_i(\xi, \theta) = (\theta_i \Psi'_i(\xi_i))^+ + \gamma_i(\xi). \quad (6)$$

As long as  $\gamma_i(\xi) = \gamma_i(\xi_i)$ , i.e. if  $\gamma_i(\xi)$  only depends on the  $i$ -th coordinate of  $\xi$ , the switching rate of coordinate  $i$  is independent of the other coordinates  $\xi_j$ ,  $j \neq i$ . It follows that the switches of the  $i$ -th coordinate can be generated by a one-dimensional time inhomogeneous Poisson process, which is independent of the switches in the other coordinates. As a consequence the  $d$ -dimensional Zig-Zag process  $(\Xi(t), \Theta(t)) = (\Xi^1(t), \dots, \Xi_d(t), \Theta^1(t), \dots, \Theta^d(t))$  is equal to the tensor product of  $d$  Zig-Zag processes  $(\Xi^i(t), \Theta^i(t))$ ,  $i = 1, \dots, d$ .

Suppose  $P(x, dy)$  is the transition kernel of a Markov chain on a state space  $E$ . We say that the Markov chain associated to  $P$  is *mixing* if there exists a probability distribution  $\pi$  on  $E$  such that

$$\lim_{k \rightarrow \infty} \|P^k(x, \cdot) - \pi\|_{\text{TV}} = 0 \quad \text{for all } x \in E.$$

For any continuous time Markov process with family of transition kernels  $P^t(x, dy)$  we can consider the associated *time-discretized process*, which is a Markov chain with transition kernel  $Q(x, dy) := P^\delta(x, dy)$  for a fixed  $\delta > 0$ . The value of  $\delta$  will be of no significance in our use of this construction.

**Proposition 2.9.** *Suppose  $\pi$  is of product form and  $\lambda : E \rightarrow \mathbb{R}_+^d$  admits the representation (6) with  $\gamma_i(\xi)$  only depending on  $\xi_i$  for  $i = 1, \dots, d$ . Furthermore suppose that for every  $i = 1, \dots, d$ , the one-dimensional time-discretized Zig-Zag process corresponding to switching rate  $\lambda_i$  is mixing in  $\mathbb{R} \times \{-1, +1\}$ . Then the time-discretized  $d$ -dimensional Zig-Zag process with switching rates  $(\lambda_i)$  is mixing. In particular, the multi-dimensional Zig-Zag process admits a unique invariant distribution in  $\mathbb{R}^d \times \{-1, +1\}^d$ .*

*Proof.* This is a direct consequence of the decomposition of the  $d$ -dimensional Zig-Zag process as  $d$  one-dimensional Zig-Zag processes and Lemma A.1, which may be found in the Appendix.  $\square$

*Example 2.10.* As a continuation of Example 2.7, consider the simple case in which  $\pi$  is of product form with each  $\pi_i$  a centered Gaussian density function with variance  $\sigma_i^2$ . It follows from Proposition 2.9 and Example 2.7 that the multi-dimensional canonical Zig-Zag process (i.e. the Zig-Zag process with  $\gamma_i \equiv 0$ ) is mixing, or more generally for any  $\gamma$  which is bounded from above and which satisfies the condition  $\gamma_i(\xi) = \gamma(\xi_i)$ . This is fundamentally different from the Bouncy Particle Sampler (Bouchard-Côté, Vollmer and Doucet (2015)), which is not ergodic for an i.i.d. Gaussian without ‘refreshments’ of the momentum variable, i.e. resampling the momentum at exponentially distributed times.

According to the following result, if the switching rates are strictly positive, the Zig-Zag process is ergodic.

**Theorem 2.11.** *Suppose  $\lambda : E \rightarrow (0, \infty)^d$ , in particular  $\lambda_i(\xi, \theta)$  is positive for all  $i = 1, \dots, d$  and  $(\xi, \theta) \in E$ . Then there exists at most a single invariant measure for the Zig-Zag process with switching rate  $\lambda$ .*

The proof of this result consists essentially of a Girsanov change of measure with respect to a Zig-Zag process targetting an i.i.d. standard normal distribution, which we know to be irreducible. The irreducibility then carries over to the Zig-Zag process with the stated switching rates. A detailed proof can be found in the Appendix.

*Remark 2.12.* Based on numerous experiments, we conjecture that the canonical multi-dimensional Zig-Zag process, i.e. with switching rates identical to zero on large parts of the state space, is ergodic in general (i.e. not only for product distributions) under only mild conditions, possibly just the stated assumptions for invariance of  $\pi$ . A detailed investigation of ergodicity of the Zig-Zag process will be the subject of a forthcoming paper.

### 3. Implementation

As mentioned earlier, the main computational challenge is an efficient simulation of the random times  $T_i^k$  introduced in Section 2.1. We will focus on simulation by means of Poisson thinning.

**Proposition 3.1** (Poisson thinning, Lewis and Shedler (1979)). *Let  $m : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  and  $M : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  be continuous such that  $m(t) \leq M(t)$  for  $t \geq 0$ . Let  $\tau^1, \tau^2, \dots$  be the increasing finite or infinite sequence of points of a Poisson process with rate function  $(M(t))_{t \geq 0}$ . For all  $i$ , delete the point  $\tau^i$  with probability  $1 - m(\tau^i)/M(\tau^i)$ . Then the remaining points  $\tilde{\tau}^1, \tilde{\tau}^2, \dots$  form a non-homogeneous Poisson process with rate function  $(m(t))_{t \geq 0}$ .*

Now for a given initial point  $(\xi, \theta) \in E$ , let  $m_i(t) := \lambda_i(\xi + \theta t, \theta)$ , for  $i = 1, \dots, d$ , and suppose we have available continuous functions  $M_i(t)$  such that  $m_i(t) \leq M_i(t)$  for  $i = 1, \dots, d$  and  $t \geq 0$ . We call these  $(M_i)_{i=1}^d$  *computational bounds* for  $(m_i)_{i=1}^d$ . We can use Proposition 3.1 to obtain the first switching times  $(\tilde{\tau}_i^1)_{i=1}^d$  from a (theoretically infinite) collection of *proposed switching times*  $(\tau_i^1, \tau_i^2, \dots)_{i=1}^d$  given the initial point  $(\xi, \theta)$ , and use the obtained skeleton point at time  $\tilde{\tau}^1 := \min_{i \in \{1, \dots, d\}} \tilde{\tau}_i^1$  as a new initial point (which is allowed by the strong Markov property) with the component  $i_0 = \operatorname{argmin}_{i \in \{1, \dots, d\}} \tilde{\tau}_i^1$  of  $\theta$  switched.

In fact, the strong Markov property of the Zig-Zag process simplifies the computational procedure even further: we can draw for each component  $i = 1, \dots, d$  the first proposed switching time  $\tau_i := \tau_i^1$ , determine  $i_0 := \operatorname{argmin}_{i \in \{1, \dots, d\}} \tau_i$  and decide whether the appropriate component of  $\theta$  is switched at this time with probability  $m_{i_0}(\tau)/M_{i_0}(\tau)$ , where  $\tau := \tau_{i_0}$ . Then since  $\tau$  is a stopping time for the Markov process, we can use the obtained point of the Zig-Zag process at

time  $\tau$  as new starting point, regardless of whether we switch a component of  $\theta$  at the obtained skeleton point. A full computational procedure for simulating the Zig-Zag process is now given by Algorithm 1.

**Algorithm 1:** Zig-Zag Sampling (ZZ)

Input: initial condition  $(\xi, \theta) \in E$ .

Output: a sequence of skeleton points  $(T^k, \Xi^k, \Theta^k)_{k=0}^\infty$ .

1.  $(T^0, \Xi^0, \Theta^0) := (0, \xi, \theta)$ .
2. for  $k = 1, 2, \dots$ 
  - (a) Define  $m_i(t) := \lambda_i(\Xi^{k-1} + \Theta^{k-1}t, \Theta^{k-1})$  for  $t \geq 0$  and  $i = 1, \dots, d$ .
  - (b) For  $i = 1, \dots, d$ , let  $(M_i)$  denote computational bounds for  $(m_i)$ .
  - (c) Draw  $\tau_1, \dots, \tau_d$  such that  $\mathbb{P}(\tau_i \geq t) = \exp\left(-\int_0^t M_i(s) ds\right)$ .
  - (d)  $i_0 := \operatorname{argmin}_{i=1, \dots, d} \{\tau_i\}$  and  $\tau := \tau_{i_0}$ .
  - (e)  $(T^k, \Xi^k) := (T^{k-1} + \tau, \Xi^{k-1} + \Theta^{k-1}\tau)$
  - (f) With probability  $m_{i_0}(\tau)/M_{i_0}(\tau)$ ,
    - $\Theta^k := F_{i_0}[\Theta^{k-1}]$ ,
    - otherwise
    - $\Theta^k := \Theta^{k-1}$ .

### 3.1. Computational bounds

We now come to the important issue of obtaining computational bounds for the Zig-Zag Proces, i.e. useful upper bounds for the switching rates  $(m_i)$ . If we can compute the inverse function  $G_i(y) := \inf\{t \geq 0 : H_i(t) \geq y\}$  of  $H_i : t \mapsto \int_0^t M_i(s) ds$ , we can simulate  $\tau_1, \dots, \tau_d$  using the CDF inversion technique, i.e. by drawing i.i.d. uniform random variables  $U_1, \dots, U_d$  and setting  $\tau_i := G_i(-\log U_i)$ ,  $i = 1, \dots, d$ .

Let us ignore the subscript  $i$  for a moment. Important examples of computational bounds are piecewise affine bounds of the form  $M : t \mapsto (a + bt)^+$ , with  $a, b \in \mathbb{R}$ , and the constant upper bounds  $M : t \mapsto c$  for  $c \geq 0$ . (Trivially, the simulated random time is  $T = \infty$  for  $M \equiv 0$ .) It is also possible to simulate using the combined rate  $M : t \mapsto \min(c, (a + bt)^+)$ . In all of these cases, the integrals  $H(t) = \int_0^t M(s) ds$  are piecewise linear or quadratic and non-decreasing, so that an explicit expression for the inverse function  $G$  can be obtained and is straightforward to implement.

The computational bounds are directly related to the algorithmic efficiency of Zig-Zag Sampling. From Algorithm 1, it is clear that for every simulated time  $\tau$  a single component of  $\lambda$  needs to be evaluated, which corresponds by (5) to the evaluation of a single component of the gradient of the negative log density  $\Psi$ . The magnitude of the computational bounds  $(M_i)$  will determine how far the Zig-Zag process will have moved in the state space before such a new evaluation

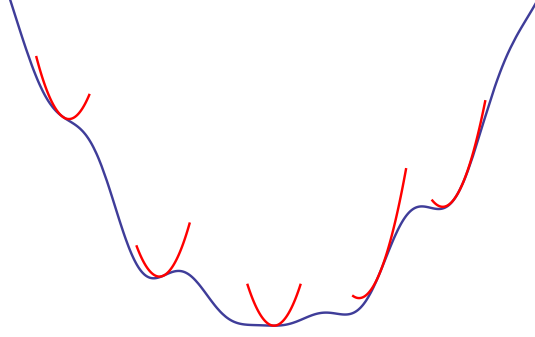


Figure 2: Illustration of a situation in which the Hessian (or second order derivative) is bounded from above. In blue is plotted an example of a negative log density, and in red fitted quadratics with matching slopes and with their curvature equal to the bound on the Hessian.

of a component of  $\lambda$  is required, and in this paper we will pay close attention to the scaling of  $M_i$  with respect to the the number of available observations in a Bayesian inference setting.

### 3.2. Example: globally bounded log density gradient

If there are constants  $c_i > 0$  such that  $\sup_{\xi \in \mathbb{R}^d} |\partial_i \Psi(\xi)| \leq c_i$ ,  $i = 1, \dots, d$ , then we can use the global upper bounds  $M_i(t) = c_i$  for  $t \geq 0$ . Indeed, for  $(\xi, \theta) \in E$ ,

$$\lambda_i(\xi, \theta) = (\theta_i \partial_i \Psi(\xi))^+ \leq |\partial_i \Psi(\xi)| \leq c_i.$$

In this case Algorithm 1 may be applied, letting  $M_i \equiv c_i$  for  $i = 1, \dots, d$  at every iteration.

This particularly simple situation arises for example with heavy-tailed distributions. E.g. if  $\pi$  is Cauchy, then  $\Psi(\xi) = \log(1 + \xi^2)$ , and consequently  $\lambda(\xi, \theta) = \left( \frac{2\theta\xi}{1+\xi^2} \right)^+ \leq 1$ .

### 3.3. Example: negative log density with dominated Hessian

A case which will often recur is the situation in which there exists a positive definite matrix  $Q \in \mathbb{R}^{d \times d}$  such that  $H_\Psi(\xi) \preceq Q$  for every  $\xi \in \mathbb{R}^d$ . Here  $H_\Psi(\xi) = (\partial_i \partial_j \Psi(\xi))_{i,j=1}^d$  denotes the Hessian matrix of  $\Psi$ . This is illustrated graphically in Figure 2. In this case we can obtain a piecewise affine computational bound, as follows.

Denote the Euclidean inner product in  $\mathbb{R}^d$  by  $\langle \cdot, \cdot \rangle$ . For  $p \in [1, \infty]$  the  $\ell^p$ -norm on  $\mathbb{R}^d$  and the induced matrix norms are both denoted by  $\|\cdot\|_p$ . For symmetric matrices  $S, T \in \mathbb{R}^{d \times d}$  we write  $S \preceq T$  if  $\langle v, Sv \rangle \leq \langle v, Tv \rangle$  for every  $v \in \mathbb{R}^d$ , or in

words, if  $T$  dominates  $S$  in the positive definite ordering. We let  $(e_i)_{i=1}^d$  denote the canonical basis vectors in  $\mathbb{R}^d$ .

For an initial value  $(\xi, \theta) \in E$ , we move along the trajectory  $t \mapsto \xi(t) := \xi + \theta t$ . Let  $a_i$  denote an upper bound for  $\theta_i \partial_i \Psi(\xi)$ ,  $i = 1, \dots, d$  and let  $b_i := \sqrt{d} \|Q e_i\|_2$ . Note that for a general symmetric matrices  $S, T$  for which  $S \preceq T$ , we have for any  $v, w \in \mathbb{R}^d$  that

$$\langle v, Sw \rangle \leq \|v\|_2 \|Sw\|_2 \leq \|v\|_2 \|Tw\|_2. \quad (7)$$

Applying this inequality we obtain for  $i = 1, \dots, d$ ,

$$\begin{aligned} \theta_i \partial_i \Psi(\xi(t)) &= \theta_i \partial_i \Psi(\xi) + \int_0^t \sum_{j=1}^d \partial_i \partial_j \Psi(\xi(s)) \theta_j \, ds \leq a_i + \int_0^t \langle H_\Psi(\xi(s)) e_i, \theta \rangle \, ds \\ &\leq a_i + \int_0^t \|Q e_i\|_2 \|\theta\|_2 \, ds = a_i + b_i t. \end{aligned}$$

It thus follows that

$$\lambda_i(\xi(t), \theta) = (\theta_i \partial_i \Psi(\xi(t)))^+ \leq (a_i + b_i t)^+.$$

Hence the general Zig-Zag Algorithm may be applied taking

$$M_i(t) := (a_i + b_i t)^+, \quad t \geq 0, \quad i = 1, \dots, d,$$

with  $a_i$  and  $b_i$  as specified above. A complete procedure for Zig-Zag Sampling for a log density with dominated Hessian is provided in Algorithm 2.

*Remark 3.2.* We could also have applied the inequality (7) to obtain the estimate

$$\langle H_\Psi(\xi(s)) e_i, \theta \rangle = \langle e_i, H_\Psi(\xi(s)) \theta \rangle \leq \|e_i\|_2 \|Q \theta\|_2,$$

which would have resulted in the choice  $b_i = \|Q \theta\|_2$ . However, the scaling of this computational bound is typically of larger magnitude: a single component of  $Q \theta$  is  $O(d)$  (assuming elements of  $Q$  are  $O(1)$ ), and therefore taking the vector norm results in a complexity  $O(d^{3/2})$ . In contrast, the size of the upper bound using  $b_i = \|Q e_i\|_2 \sqrt{d}$  is only  $O(d)$ .

#### 4. Big data Bayesian inference by means of error-free sub-sampling

Throughout this section we assume the derivatives of  $\Psi$  admit the representation

$$\partial_i \Psi(\xi) = \frac{1}{n} \sum_{j=1}^n E_i^j(\xi), \quad i = 1, \dots, d, \quad \xi \in \mathbb{R}^d, \quad (8)$$

with  $(E^j)_{j=1}^n$  functions in  $C(\mathbb{R}^d, \mathbb{R}^d)$ . The motivation for considering such a class of density functions is the problem of sampling from a posterior distribution for big data. The key feature of such posteriors is that they can be written as the



**Algorithm 2:** Zig-Zag Sampling for log density with dominated HessianInput: initial condition  $(\xi, \theta) \in E$ .Output: a sequence of skeleton points  $(T^k, \Xi^k, \Theta^k)_{k=0}^\infty$ .

1.  $(T^0, \Xi^0, \Theta^0) := (0, \xi, \theta)$ .
2.  $a_i := \theta_i \partial_i \Psi(\xi)$ ,  $i = 1, \dots, d$ .
3.  $b_i := Q e_i \sqrt{d}$ ,  $i = 1, \dots, d$ .
4. For  $k = 1, 2, \dots$ 
  - (a) Draw  $\tau_i$  such that  $\mathbb{P}(\tau_i \geq t) = \exp\left(-\int_0^t (a_i + b_i s)^+ ds\right)$ ,  $i = 1, \dots, d$ .
  - (b)  $i_0 := \operatorname{argmin}_{i \in \{1, \dots, d\}} \tau_i$  and  $\tau := \tau_{i_0}$ .
  - (c)  $(T^k, \Xi^k, \Theta^k) := (T^{k-1} + \tau, \Xi^{k-1} + \Theta^{k-1} \tau, \Theta^{k-1})$
  - (d)  $a_i := a_i + b_i \tau$ ,  $i = 1, \dots, d$ .
  - (e) with probability  $\frac{(\Theta_{i_0}^{k-1} \partial_{i_0} \Psi(\Xi^k))^+}{(a_{i_0})^+}$ ,
    - $\Theta^k := F_{i_0}[\Theta^{k-1}]$
otherwise
    - $\Theta^k := \Theta^{k-1}$ .
  - (f)  $a_{i_0} := \Theta_{i_0}^{k-1} \partial_{i_0} \Psi(\Xi^k)$  (re-using the earlier computation)

product of a large number of terms. For example consider the simplest example of this, where we have  $n$  independent data points and for which the likelihood function is

$$L(x^{1:n}|\xi) = \prod_{j=1}^n f(x^j|\xi),$$

for some probability density or probability mass function  $f$ . In this case we can write the negative log density  $\Psi$  associated with the posterior distribution as an average

$$\Psi(\xi) = \frac{1}{n} \sum_{j=1}^n \Psi^j(\xi), \quad (9)$$

where  $\Psi^j(\xi) = -\log \pi_0(\xi) - n \log f(x^j|\xi)$ , and we could choose  $E_i^j(\xi) = \partial_i \Psi^j(\xi)$ . It is crucial that every  $E_i^j$  is a factor  $O(n)$  cheaper to evaluate than the full derivative  $\partial_i \Psi(\xi)$ .

We will describe two successive improvements over the basic Zig-Zag Sampling (ZZ) algorithm specifically tailored to the situation in which (8) is satisfied. The first improvement consists of a sub-sampling approach that means we need to calculate only one of the  $E_i^j$ s at each simulated time, rather than sum of all  $n$  of these quantities. This sub-sampling approach (referred to as Zig-Zag with Sub-Sampling, ZZ-SS) comes at the cost of an increased computational bound. Our second improvement is to use control variate ideas to reduce this bound, resulting in the Zig-Zag with Control Variates (ZZ-CV) algorithm.

#### 4.1. Main idea

Let  $(\xi(t))_{t \geq 0}$  denote a linear trajectory originating in  $(\xi, \theta) \in E$ , i.e.  $\xi(t) = \xi + \theta t$ . Define a collection of switching rates along the trajectory  $(\xi(t))$  by

$$m_i^j(t) := \left( \theta_i E_i^j(\xi(t)) \right)^+, \quad i = 1, \dots, d, \quad j = 1, \dots, n, \quad t \geq 0.$$

We will make use of computational bounds  $(M_i)$  as before, which this time bound  $(m_i^j)$  uniformly. More specifically, let  $M_i : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  be continuous and satisfy

$$m_i^j(t) \leq M_i(t) \quad \text{for all } i = 1, \dots, d, j = 1, \dots, n, \text{ and } t \geq 0. \quad (10)$$

We will generate random times according to the computational upper bounds  $(M_i)$  as before. However, we now use a two-step approach to deciding whether to switch or not at the generated times. As before, for  $i = 1, \dots, d$  let  $(\tau_i)_{i=1}^d$  be simulated random times for which  $\mathbb{P}(\tau \geq t) = \exp\left(-\int_0^t M_i(s) ds\right)$  and let  $i_0 := \operatorname{argmin}_{i \in \{1, \dots, d\}} \tau_i$ , and  $\tau := \tau_{i_0}$ . Then switch component  $i_0$  of  $\theta$  with probability  $m_{i_0}^J(\tau)/M_{i_0}(\tau)$ , where  $J \in \{1, \dots, n\}$  is drawn uniformly at random, independent of  $\tau$ . This ‘sub-sampling’ procedure is provided in pseudo-code in Algorithm 3. Depending on the choice of  $E_i^j$ , we will refer to his algorithm as Zig-Zag with Sub-Sampling (ZZ-SS, Section 4.2) or ZZ-CV (Section 4.3).

**Theorem 4.1.** *Algorithm 3 generates a skeleton of a Zig-Zag process with switching rates given by*

$$\lambda_i(\xi, \theta) = \frac{1}{n} \sum_{j=1}^n \left( \theta_i E_i^j(\xi) \right)^+, \quad i = 1, \dots, d, \quad (\xi, \theta) \in E, \quad (11)$$

and invariant distribution  $\mu$  given by (4).

*Proof.* Conditional on  $\tau$ , the probability that component  $i_0$  of  $\theta$  is switched at time  $\tau$  is seen to be

$$\mathbb{E}_J \left[ m_{i_0}^J(\tau) / M_{i_0}(\tau) \right] = \frac{\frac{1}{n} \sum_{j=1}^n m_{i_0}^j(\tau)}{M_{i_0}(\tau)} = \frac{m_{i_0}(\tau)}{M_{i_0}(\tau)},$$

where

$$m_i(t) := \frac{1}{n} \sum_{j=1}^n m_i^j(t) = \frac{1}{n} \sum_{j=1}^n \left( \theta_i E_i^j(\xi(t)) \right)^+, \quad i = 1, \dots, d, \quad t \geq 0.$$

By Proposition 3.1 we thus have an effective switching rate  $\lambda_i$  for switching the  $i$ -th component of  $\theta$  given by (11). Finally we verify that the switching rates  $(\lambda_i)$  given by (11) satisfy (3). Indeed,

$$\begin{aligned} \lambda_i(\xi, \theta) - \lambda_i(\xi, F_i[\theta]) &= \frac{1}{n} \sum_{j=1}^n \left\{ \left( \theta_i E_i^j(\xi) \right)^+ - \left( \theta_i E_i^j(\xi) \right)^- \right\} \\ &= \frac{1}{n} \sum_{j=1}^n \theta_i E_i^j(\xi) = \theta_i \partial_i \Psi(\xi). \end{aligned}$$

By Theorem 2.2, it follows that the associated Zig-Zag process has the stated invariant distribution.  $\square$

**Algorithm 3:** Zig-Zag with Sub-Sampling (ZZ-SS) / Zig-Zag with Control Variates (ZZ-CV)

Input: initial condition  $(\xi, \theta) \in E$ .  
Output: a sequence of skeleton points  $(T^k, \Xi^k, \Theta^k)_{k=0}^\infty$ .

1.  $(T^0, \Xi^0, \Theta^0) := (0, \xi, \theta)$ .
2. for  $k = 1, 2, \dots$ 
  - (a) Define  $m_i^j(t) := \left( \Theta^{k-1} E_i^j(\Xi^{k-1} + \Theta^{k-1}t) \right)^+$  for  $t \geq 0$ ,  $i = 1, \dots, d$  and  $j = 1, \dots, n$ .
  - (b) For  $i = 1, \dots, d$ , let  $(M_i)$  denote computational bounds for  $(m_i^j)$ , i.e. satisfying (10).
  - (c) Draw  $\tau_1, \dots, \tau_d$  such that  $\mathbb{P}(\tau_i \geq t) = \exp\left(-\int_0^t M_i(s) ds\right)$ .
  - (d)  $i_0 := \operatorname{argmin}_{i=1, \dots, d} \tau_i$  and  $\tau := \tau_{i_0}$ .
  - (e)  $(T^k, \Xi^k) := (T^{k-1} + \tau, \Xi^{k-1} + \Theta^{k-1}\tau)$
  - (f) Draw  $J \sim \text{Uniform}(\{1, \dots, n\})$ .
  - (g) With probability  $m_{i_0}^J(\tau)/M_{i_0}(\tau)$ ,
    - $\Theta^k := F_{i_0}[\Theta^{k-1}]$ ,
otherwise
    - $\Theta^k := \Theta^{k-1}$ .

The important advantage of using Zig-Zag in combination with sub-sampling is that at every iteration of the algorithm we only have to evaluate a single component of  $E_i^j$ , which reduces algorithmic complexity by a factor  $O(n)$ . However this may come at a cost. Firstly, the computational bounds  $(M_i)$  may have to be increased which in turn will increase the algorithmic complexity of simulating the Zig-Zag sampler. Also, the dynamics of the Zig-Zag process will change, because the actual switching rates of the process are increased. This increases the diffusivity of the continuous time Markov process, and affects the mixing properties in a negative way.

#### 4.2. Zig-Zag with Sub-Sampling (ZZ-SS) for globally bounded log density gradient

A straightforward application of sub-sampling is possible if we can write

$$\Psi(\xi) = \frac{1}{n} \sum_{j=1}^n \Psi^j(\xi), \quad \xi \in \mathbb{R}^d, \quad (12)$$

such that  $\nabla\Psi^j$  are globally bounded, i.e. there exist positive constants  $(c_i)$  such that

$$|\partial_i\Psi^j(\xi)| \leq c_i, \quad i = 1, \dots, d, \quad j = 1, \dots, n, \quad \xi \in \mathbb{R}^d. \quad (13)$$

In this case we may take

$$E_i^j := \partial_i\Psi^j \quad \text{and} \quad M_i(t) := c_i, \quad i = 1, \dots, d, \quad j = 1, \dots, n \quad t \geq 0,$$

so that (10) is satisfied. The corresponding version of Algorithm 3 will be called Zig-Zag with Sub-Sampling (ZZ-SS).

#### 4.3. Zig-Zag with Control Variates (ZZ-CV)

Suppose again that  $\Psi$  admits the representation (12), and further suppose that the derivatives  $(\partial_i\Psi^j)$  are globally and uniformly Lipschitz, i.e., there exist constants  $(C_i)_{i=1}^n$  such that for some  $p \in [1, \infty]$  and all  $i = 1, \dots, d, j = 1, \dots, n$ , and  $\xi_1, \xi_2 \in \mathbb{R}^d$ ,

$$|\partial_i\Psi^j(\xi_1) - \partial_i\Psi^j(\xi_2)| \leq C_i \|\xi_1 - \xi_2\|_p. \quad (14)$$

To use these Lipschitz bounds we need to choose a reference point  $\xi^*$  in  $\xi$ -space, so that we can bound the derivative of the log density based on how close we are to this reference point. Now if we choose any fixed reference point,  $\xi^* \in \mathbb{R}^d$ , we can use a control variate idea to write

$$\partial_i\Psi(\xi) = \partial_i\Psi(\xi^*) + \frac{1}{n} \sum_{i=1}^n [\partial_i\Psi^j(\xi) - \partial_i\Psi^j(\xi^*)], \quad \xi \in \mathbb{R}^d, \quad i = 1, \dots, d.$$

This suggests using

$$E_i^j(\xi) := \partial_i\Psi(\xi^*) + \partial_i\Psi^j(\xi) - \partial_i\Psi^j(\xi^*), \quad \xi \in \mathbb{R}^d, \quad i = 1, \dots, d, \quad j = 1, \dots, n.$$

The reason for defining  $E_i^j(\xi)$  in this manner is to try and reduce the variability of the value of these terms as we vary  $j$ . By the Lipschitz condition we have  $E_i^j(\xi) \leq C_i \|\xi - \xi^*\|_p$ , and thus the variability of the  $E_i^j(\xi)$ s will be small if  $\xi$  is close to  $\xi^*$ . The first term for  $E_i^j$  suggests the reference point  $\xi^*$  should be close to the mode of the posterior. Under standard asymptotics we expect a draw from the posterior for  $\xi$  to be  $O_p(n^{-1/2})$  from the posterior mode. Thus if we have a procedure for finding a reference point  $\xi^*$  which is within  $O(n^{-1/2})$  of the posterior mode then this would ensure  $\|\xi - \xi^*\|_2$  is  $O(n^{-1/2})$  if  $\xi$  is drawn from the posterior. For such a choice of  $\xi^*$  we would have  $\partial_i\Psi(\xi^*)$  of  $O_p(n^{1/2})$ .

Using the Lipschitz condition, we can now obtain computational bounds of  $(m_i)$  for a trajectory  $\xi(t) := \xi + \theta t$  originating in  $(\xi, \theta)$ . Define

$$M_i(t) := a_i + b_i t, \quad t \geq 0, \quad i = 1, \dots, d,$$

where  $a_i := (\theta_i \partial_i \Psi(\xi^*))^+ + C_i \|\xi - \xi^*\|_p$  and  $b_i := C_i d^{1/p}$ . Then (10) is satisfied. Indeed, using Lipschitz continuity of  $y \mapsto (y)^+$ ,

$$\begin{aligned} m_i^j(t) &= \left( \theta_i E_i^j(\xi + \theta t) \right)^+ = (\theta_i \partial_i \Psi(\xi^*) + \theta_i \partial_i \Psi^j(\xi + \theta t) - \theta_i \partial_i \Psi^j(\xi^*))^+ \\ &\leq (\theta_i \partial_i \Psi(\xi^*))^+ + |\partial_i \Psi^j(\xi) - \partial_i \Psi^j(\xi^*)| + |\partial_i \Psi^j(\xi + \theta t) - \partial_i \Psi^j(\xi)| \\ &\leq (\theta_i \partial_i \Psi(\xi^*))^+ + C_i (\|\xi - \xi^*\|_p + t \|\theta\|_p) = M_i(t). \end{aligned}$$

Implementing this scheme requires some pre-processing of the data. First we need a way of choosing a suitable reference point  $\xi^*$  to find a value close of the mode using an approximate or exact numerical optimization routine. The complexity of this operation will be  $O(n)$ . Once we have found such a reference point we have an one-off  $O(n)$  cost of calculating  $\partial_i \Psi(\xi^*)$  for each  $i = 1, \dots, d$ . However, once we have paid this upfront computational cost, the resulting Zig-Zag sampler can be super-efficient. This is discussed in more detail in Section 5, and demonstrated empirically in Section 6. The version of Algorithm 3 resulting from this choice of  $E_i^j$  and  $M_i$  will be called Zig-Zag with Control Variates (ZZ-CV).

*Remark 4.2* (Choice of  $p$ ). When choosing  $p \geq 1$ , in general there will be a trade-off between the magnitude of  $C_i$  and of  $\|\xi - \xi^*\|_p$ , which may influence the scaling of Zig-Zag sampling with dimension. For example, we will see in Section 6.4 that for i.i.d. Gaussian components, the choice  $p = \infty$  is optimal. When the situation is not so clear, choosing the Euclidean norm ( $p = 2$ ) is as reasonable as any other choice.

## 5. Scaling analysis

In this section we provide an informal argument for how (i) Canonical Zig-Zag, and (ii) Zig-Zag with control variates and sub-sampling, behave for big data.

For the moment we fix  $n \in \mathbb{N}$ . We will consider a posterior with negative log density

$$\Psi(\xi) = - \sum_{j=1}^n \log f(x^j \mid \xi),$$

where  $x^j$  are i.i.d. drawn from  $f(x^j \mid \xi_0)$ . Let  $\hat{\xi}$  denote the maximum likelihood estimator (MLE) for  $\xi$  based on data  $x^1, \dots, x^n$ . Introduce the coordinate transformation

$$\phi(\xi) = \sqrt{n}(\xi - \hat{\xi}), \quad \xi(\phi) = \frac{1}{\sqrt{n}}\phi + \hat{\xi}.$$

As  $n \rightarrow \infty$  the posterior distribution in terms of  $\phi$  will converge to a multivariate Gaussian distribution with mean 0 and covariance matrix given by the inverse of the expected information  $i(\theta_0)$ ; see e.g. Johnson (1970).

### 5.1. Scaling of Zig-Zag Sampling (ZZ)

First let us obtain a Taylor expansion of the switching rate for  $\xi$  close to  $\hat{\xi}$ . We have

$$\begin{aligned}\partial_{\xi_i} \Psi(\xi) &= -\partial_{\xi_i} \sum_{j=1}^n \log f(x^j \mid \xi) \\ &= \underbrace{-\partial_{\xi_i} \sum_{j=1}^n \log f(x^j \mid \hat{\xi})}_{=0} - \sum_{j=1}^n \sum_{k=1}^d \partial_{\xi_i} \partial_{\xi_k} \log f(x^j \mid \hat{\xi})(\xi_k - \hat{\xi}_k) + O(\|\xi - \hat{\xi}\|^2).\end{aligned}$$

The first term vanishes by the definition of the MLE. Expressed in terms of  $\phi$ , the switching rates are

$$(\theta_i \partial_{\xi_i} \Psi(\xi(\phi)))^+ = \frac{1}{\sqrt{n}} \underbrace{\left( -\sum_{j=1}^n \sum_{k=1}^d \partial_{\xi_i} \partial_{\xi_k} \log f(x^j \mid \hat{\xi}) \phi_k \right)^+}_{O(\sqrt{n})} + O\left(\frac{\|\phi\|^2}{n}\right).$$

With respect to the coordinate  $\phi$ , the canonical Zig-Zag process has constant speed  $\sqrt{n}$  in each coordinate, and by the above computation, a switching rate of  $O(\sqrt{n})$ . After a rescaling of the time parameter by a factor  $\sqrt{n}$ , the process in the  $\phi$ -coordinate becomes a Zig Zag process with unit speed in every direction and switching rates

$$\left( -\frac{1}{n} \sum_{j=1}^n \sum_{k=1}^d \partial_{\xi_i} \partial_{\xi_k} \log f(x^j \mid \xi) \phi_k \right)^+ + O(n^{-1/2}).$$

If we let  $n \rightarrow \infty$ , the switching rates converge almost surely to those of a Zig Zag process with switching rates

$$\tilde{\lambda}_i(\phi, \theta) = (\theta_i (i(\theta_0) \phi)_i)^+$$

where  $i(\theta_0)$  denotes the expected information. These switching rates correspond to the limiting Gaussian distribution with covariance matrix  $(i(\theta_0))^{-1}$ .

In this limiting Zig-Zag process, all dependence on  $n$  has vanished. Starting from equilibrium, we require a time interval of  $O(1)$  (in the rescaled time) to obtain an essentially independent sample. Going back to the original time scale, this corresponds to a time interval of  $O(n^{-1/2})$ . As long as the computational bound in the Zig-Zag algorithm is  $O(n^{1/2})$ , this can be achieved using  $O(1)$  proposed switches. The computational cost for every proposed switch is  $O(n)$ , because the full data  $(x^i)_{i=1}^n$  needs to be processed in the computation of the true switching rate at the proposed switching time.

*We conclude that the computational complexity of the Zig-Zag (ZZ) algorithm per independent sample is  $O(n)$ , provided that the computational bound is*

$O(n^{1/2})$ . This is the best we can expect for any standard Monte Carlo algorithm (where we will have a  $O(1)$  number of iterations, but each iteration is  $O(n)$  in computational cost).

To compare, if the computational bound is  $O(n^\alpha)$  for some  $\alpha > 1/2$ , then we require  $O(n^{\alpha-1/2})$  proposed switches before we have simulated a total time interval of length  $O(n^{-1/2})$ , so that, with a complexity of  $O(n)$  per proposed switching time, the Zig-Zag algorithm has total computational complexity  $O(n^{\alpha+1/2})$ . So, for example, with global bounds we have that the computational bound is  $O(n)$  (as each term in the log density is  $O(1)$ ), and hence ZZ will have total computational complexity of  $O(n^{3/2})$ .

*Example 5.1 (Dominated Hessian).* Consider Algorithm 2 in the one-dimensional case, with the second derivative of  $\Psi$  bounded from above by  $Q > 0$ . We have  $Q = O(n)$  as  $\Psi''$  is the sum of  $n$  terms of  $O(1)$ . The value of  $b$  is kept fixed at the value  $b = Q = O(n)$ . Next  $a$  is given initially as

$$a = \theta\Psi'(\xi) \leq \underbrace{\theta\Psi'(\hat{\xi})}_{=0} + \underbrace{Q}_{O(n)} \underbrace{(\xi - \hat{\xi})}_{O(n^{-1/2})} = O(n^{1/2}),$$

and increased by  $b\tau$  until a switch happens and  $a$  is reset to  $\theta\Psi'(\xi)$ . Because of the initial value for  $a$ , switches will occur at rate  $O(n^{1/2})$  so that  $\tau$  will be  $O(n^{-1/2})$ , and the value of  $a$  will remain  $O(n^{1/2})$ . Hence the magnitude of the computational bound  $M(t) = (a + bt)^+$  is  $O(n^{1/2})$ .

## 5.2. Scaling of Zig-Zag with Control Variates (ZZ-CV)

Now we will study the limiting behaviour as  $n \rightarrow \infty$  of ZZ-CV introduced in Section 4.3. In determining the computational bounds we take  $p = 2$  for simplicity, e.g. in (14). Also for simplicity assume that  $\xi \mapsto \partial_{\xi_i} \log f(x^j | \xi)$  has Lipschitz constant  $k_i$  (independent of  $j = 1, \dots, n$ ) and write  $C_i = nk_i$ , so that (14) is satisfied. In practice there may be a logarithmic increase with  $n$  in the Lipschitz constants  $k_i$  as we have to take a global bound in  $n$ , see e.g. (17) in Section 6.5. For the present discussion we ignore such logarithmic factors. We assume reference points  $\xi^*$  for growing  $n$  are determined in such a way that  $\|\xi^* - \hat{\xi}\|_2$  is  $O(n^{-1/2})$ . For definiteness, suppose there exists a  $d$ -dimensional random variable  $Z$  such that  $n^{1/2}(\xi^* - \hat{\xi}) \rightarrow Z$  in distribution, with the randomness in  $Z$  independent of  $(x^j)_{j=1}^\infty$ .

We can look at CV-ZZ with respect to the scaled coordinate  $\phi$  as  $n \rightarrow \infty$ . Denote the reference point for the rescaled parameter as  $\phi^* := \sqrt{n}(\xi^* - \hat{\xi})$ .

The essential quantities to consider are the switching rate estimators  $E_i^j$ . We estimate

$$\begin{aligned} |E_i^j(\xi)| &= |\partial_{\xi_i} \Psi(\xi^*) + \partial_{\xi_i} \Psi^j(\xi) - \partial_{\xi_i} \Psi^j(\xi^*)| \\ &= |\partial_{\xi_i} \Psi(\xi^*) - \partial_{\xi_i} \Psi(\hat{\xi}) + \partial_{\xi_i} \Psi^j(\xi) - \partial_{\xi_i} \Psi^j(\xi^*)| \\ &\leq \underbrace{C_i}_{O(n)} \underbrace{\|\xi^* - \hat{\xi}\|}_{O(n^{-1/2})} + \underbrace{C_i}_{O(n)} \underbrace{\|\xi - \xi^*\|}_{O(n^{-1/2})}. \end{aligned}$$

We find that  $|E_i^j(\xi)| = O(n^{1/2})$  under the stationary distribution.

By slowing down the Zig-Zag process in  $\phi$  space by  $\sqrt{n}$ , the continuous time process generated by ZZ-CV will approach a limiting Zig-Zag process with a certain switching rate of  $O(1)$ . In general this switching rate will depend on the way that  $\xi^*$  is obtained. To simplify the exposition, in the following computation we assume  $\xi^* = \hat{\xi}$ . Rescaling by  $n^{-1/2}$ , and developing a Taylor approximation around  $\hat{\xi}$ ,

$$\begin{aligned} n^{-1/2} E_i^j(\xi) &= n^{-1/2} \left( \partial_{\xi_i} \Psi^j(\xi) - \partial_{\xi_i} \Psi^j(\hat{\xi}) \right) \\ &= n^{-1/2} \left( -n \partial_{\xi_i} \log f(x^j | \xi) + n \partial_{\xi_i} \log f(x^j | \hat{\xi}) \right) \\ &= -n^{1/2} \left( \sum_{k=1}^d \partial_{\xi_i} \partial_{\xi_k} \log f(x^j | \hat{\xi}) (\xi_k - \hat{\xi}_k) \right) + O(n^{1/2} \|\xi - \hat{\xi}\|^2) \\ &= - \sum_{k=1}^d \partial_{\xi_i} \partial_{\xi_k} \log f(x^j | \hat{\xi}) \phi_k + O(n^{-1/2}). \end{aligned}$$

By Theorem 4.1, the rescaled effective switching rate for ZZ-CV is given by

$$\begin{aligned} \tilde{\lambda}_i(\phi, \theta) &:= n^{-1/2} \lambda_i(\xi(\phi), \theta) = \frac{1}{n^{3/2}} \sum_{j=1}^n \left( \theta_i E_i^j(\xi(\phi)) \right)^+ \\ &= \frac{1}{n} \sum_{j=1}^n \left( -\theta_i \sum_{k=1}^d \partial_{\xi_i} \partial_{\xi_k} \log f(x^j | \hat{\xi}) \phi_k \right)^+ + O(n^{-1/2}) \\ &\rightarrow \mathbb{E} \left( -\theta_i \sum_{k=1}^d \partial_{\xi_i} \partial_{\xi_k} \log f(X | \xi_0) \phi_k \right)^+, \end{aligned}$$

where  $\mathbb{E}$  denotes expectation with respect to  $X$ , with density  $f(\cdot | \xi_0)$ , and the convergence is a consequence of the law of large numbers. If  $\xi^*$  is not exactly equal to  $\hat{\xi}$ , the limiting form of  $\tilde{\lambda}_i(\phi, \theta)$  will be different, but the important point is that it will be  $O(1)$ , which follows from the bound on  $|E_i^j|$  above.

Just as with ZZ, the rescaled Zig-Zag process underlying ZZ-CV converges to a limiting Zig-Zag process with switching rate  $\tilde{\lambda}_i(\phi, \theta)$ . Since the computational bounds of ZZ-CV are  $O(n^{1/2})$ , a completely analogous reasoning to the one for ZZ algorithm above (Section 5.1) leads to the conclusion that  $O(1)$  proposed switches are required to obtain an independent sample. However, in contrast with the ZZ-algorithm, the ZZ-CV algorithm is designed in such a way that the computational cost per proposed switch is  $O(1)$ .

*We conclude that the computational complexity of the ZZ-CV algorithm is  $O(1)$  per independent sample. This provides a factor  $n$  increase in efficiency over standard MCMC algorithms, resulting in an unbiased algorithm for which the computational cost of obtaining an independent sample does not depend on the size of the data.*



### 5.3. Remarks

The arguments above assume we are at stationarity – and how quickly the two algorithms converge is not immediately clear. Note however that for sub-sampling Zig-Zag it is possible to choose the reference point  $\xi^*$  as starting point, thus avoiding much of the issues about convergence.

In some sense, the good computational scaling of ZZ-CV is leveraging the asymptotic normality of the posterior, but in such a way that ZZ-CV always samples from the true posterior. Thus when the posterior is close to Gaussian it will be quick; when it is far from Gaussian it may well be slower but will still be “correct”. This is fundamentally different from other algorithms (e.g. Neiswanger, Wang and Xing, 2013; Scott, Blocker and Bonassi, 2016) that utilise the asymptotic normality in terms of justifying their approximation to the posterior. Such algorithms are accurate if the posterior is close to Gaussian, but may be inaccurate otherwise, and it is often impossible to quantify the size of the approximation in practice.

## 6. Examples and experiments

### 6.1. Sampling and integration along Zig-Zag trajectories

There are essentially two different ways of using the Zig-Zag skeleton points which we obtain by using e.g. Algorithms 1, 2, or 3.

The first possible approach is to collect a number of samples along the trajectories. For, this suppose we have simulated the Zig-Zag process up to time  $\tau > 0$ , and we wish to collect  $k$  samples. This can be achieved by setting  $t_i = i\tau/m$ , and setting  $\Xi_i := \Xi(t_i)$  for  $i = 1, \dots, m$ , with the continuous time trajectory  $(\Xi(t))$  defined as in Section 2.1. This approach offers a straightforward way to compare with discrete time MCMC algorithms. Effectively, the continuous time Zig-Zag process determined by the family of transition kernels  $P_t((\xi, \theta), \cdot)$  is transformed into a discrete time Markov chain with transition kernel  $P_{t_1}((\xi, \theta), \cdot)$ . In order to approximate  $\pi(f)$  numerically for some function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  of interest, we can use the usual ergodic average

$$\widehat{\pi(f)} := \frac{1}{m} \sum_{i=1}^m f(\Xi_i).$$

An issue with this approach is that we have to decide on the amount of samples we wish to use. Taking the number of samples of the same order as the number of switches made along the Zig-Zag trajectory is a good rule of thumb.

It is important that one does not make the mistake of using the switching points of the Zig Zag process as samples, as these points are not distributed according to  $\pi$ . In particular, the switching points are biased towards the tails of the target distribution.

An alternative approach is intrinsically related to the continuous time and piecewise linear nature of the Zig-Zag trajectories. This approach consists of

continuous time integration of the Zig-Zag process, which can in many cases be performed exactly. By the continuous time ergodic theorem, for  $f$  as above,  $\pi(f)$  can be estimated as

$$\widehat{\pi(f)} = \frac{1}{\tau} \int_0^\tau f(\Xi(s)) \, ds.$$

Since the output of the Zig-Zag algorithms consists of a finite number of skeleton points  $(T^i, \Xi^i, \Theta^i)_{i=0}^k$ , we can express this as

$$\widehat{\pi(f)} = \frac{1}{T^k} \sum_{i=1}^k \int_{T^{i-1}}^{T^i} f(\Xi^{i-1} + \Theta^{i-1}(s - T^{i-1})) \, ds.$$

Due to the piecewise linearity of  $\Xi(t)$ , in many cases these integrals can be computed exactly, e.g. for the moments,  $f(x) = x^p$ ,  $p \in \mathbb{R}$ . In cases where the integral can not be computed exactly, numerical quadrature rules can be applied. An advantage of this method is that we do not have to make an arbitrary decision on the number of samples to extract from the trajectory.

## 6.2. Effective Sample Size for continuous time trajectories

In order to perform numerical experiments we compute, for an obtained continuous time Zig-Zag trajectory  $(\Xi(t), \Theta(t))$ , the associated Effective Sample Size (ESS) corresponding to a continuous observable  $h : \mathbb{R}^d \rightarrow \mathbb{R}$ . We say that the Central Limit Theorem (CLT) holds for  $(h(\Xi(t)))_{t \geq 0}$  if, as  $t \rightarrow \infty$ , the distribution of

$$\frac{1}{\sqrt{t}} \int_0^t \{h(\Xi(s)) - \pi(h)\} \, ds$$

converges in distribution to a centred normal distribution with variance  $\sigma_h^2$ , called the *asymptotic variance*. The asymptotic variance can be estimated by dividing an obtained trajectory  $(\Xi(t))_{0 \leq t \leq \tau}$  into  $B$  intervals (“batches”) of length  $\tau/B$ . Under the assumption that the batch are sufficiently large, we have that

$$Y_i := \sqrt{\frac{B}{\tau}} \int_{(i-1)\tau/B}^{i\tau/B} h(\Xi(s)) \, ds$$

has approximately a  $N(\sqrt{\frac{\tau}{B}}\pi(h), \sigma_h^2)$  distribution, for  $i = 1, \dots, B$ . Making the further approximating assumption that the random variables  $(Y_i)$  are independent (which is reasonable if the batches themselves are sufficiently long), we may estimate  $\sigma_h^2$  as the sample variance of  $(Y_i)_{i=1, \dots, B}$ , i.e. we use the estimator

$$\widehat{\sigma_h^2} = \frac{1}{B-1} \sum_{i=1}^B (Y_i - \bar{Y})^2,$$

with  $\bar{Y} = \frac{1}{B} \sum_{i=1}^B Y_i$ . We also estimate the mean and variance of  $h$  under  $\pi$  by

$$\widehat{\pi(h)} := \frac{1}{\tau} \int_0^\tau h(\Xi(s)) \, ds, \quad \widehat{\text{Var}_\pi h} := \frac{1}{\tau} \int_0^\tau h(\Xi(s))^2 \, ds - \left(\widehat{\pi(h)}\right)^2,$$

which converge almost surely as  $\tau \rightarrow \infty$  to the true mean and variance under the condition that the Zig-Zag process is ergodic. The estimate for Effective Sample Size is now given as

$$\widehat{ESS} := \frac{\tau \widehat{\text{Var}}_{\pi}(h)}{\widehat{\sigma}_h^2}.$$

### 6.3. Beating one ESS per epoch

We use the term *epoch* as a unit of computational cost, corresponding to the number of iterations required to evaluate the complete gradient of  $\log \pi$ . This means that for the basic Zig-Zag algorithm (without sub-sampling), an epoch consists of exactly one iteration, and for the sub-sampled variants of the Zig-Zag algorithm, an epoch consists of  $n$  iterations. The CPU running times per epoch of the various algorithms we consider are equal up to a constant factor. To assess the scaling of various algorithms, we use *ESS per epoch*. Consider any classical MCMC algorithm based upon the Metropolis-Hastings acceptance rule. Since every iteration requires an evaluation of the full density function to compute the acceptance probability, we have that the ESS per epoch for such an algorithm is bounded from above by one. Similar observations apply to all other known MCMC algorithms capable of sampling asymptotically from the exact target distribution.

There do exist several conceptual innovations based on the idea of sub-sampling, which have some theoretical potential to overcome the fundamental limitation of one ESS per epoch sketched above. We will briefly discuss the two most prominent examples of such methods: the Pseudo-Marginal Method (PMM, [Andrieu and Roberts \(2009\)](#)), and Stochastic Gradient Langevin Dynamics (SGLD, [Welling and Teh \(2011\)](#)), which we will consider in more detail.

The PMM is based upon using a positive unbiased estimator for a possibly unnormalized density. Obtaining an unbiased estimator of a product is much more difficult than obtaining one for a sum. Furthermore, it has been shown to be impossible to construct an estimator that is guaranteed to be positive without other information about the product, such as a bound on the terms in the product [Jacob and Thiery \(2015\)](#). Therefore the PMM does not apply in a straightforward way to vanilla MCMC in Bayesian inference and we will not consider the PMM further here.

#### 6.3.1. Scaling of Stochastic Gradient Langevin Dynamics for large datasets

For notational simplicity we will focus on a 1-dimensional target, though the arguments below apply more generally. The SGLD algorithm consists of stochastic updates of the form

$$\Xi_i := \Xi_{i-1} + \frac{1}{2} h_i \widehat{\nabla_{\xi} \log \pi}(\Xi_{i-1}) + \sqrt{h_i} Z_i, \quad (15)$$

where  $(h_i)$  is a sequence of positive step sizes,  $(Z_i)$  are independent  $N(0, 1)$  random variables, and where  $\widehat{\log \pi}(\xi)$  is an unbiased estimator of  $\log \pi(\xi)$  for  $\xi \in \mathbb{R}^d$ . In practice,  $\widehat{\nabla \log \pi}$  will be constructed using randomly sampled batches of fixed size  $m \in \{1, \dots, n\}$ ,

$$\widehat{\nabla \log \pi}(\xi) := \frac{n}{m} \sum_{i=1}^m \nabla_{\xi} \left( \frac{1}{n} \log \pi_0(\xi) + f(x^{J_i} \mid \xi) \right),$$

where  $(J_i)_{i=1}^m$  are drawn uniformly without replacement from  $\{1, \dots, n\}$ . Under certain conditions, in particular on the decay of the step sizes to 0 as  $i \rightarrow \infty$ , SGLD provides an asymptotically unbiased approximation of the target distribution  $\pi$ ; see [Teh, Thiery and Vollmer \(2014\)](#) for a detailed analysis. However the Monte Carlo error of the resulting algorithm decays at a slower rate than for standard MCMC algorithms.

As in Section 3 it is natural to study the behaviour of SGLD for a scaled variable,  $\phi(\xi) := \sqrt{n}(\xi - \hat{\xi})$ , that converges to a fixed distribution as  $n \rightarrow \infty$ . With respect to the reparametrization  $\phi$ , the updates (15) correspond to

$$\Phi_i := \Phi_{i-1} + \frac{1}{2} h n \widehat{\nabla_{\phi} \log \pi}(\Phi_{i-1}) + \sqrt{h n} Z_i,$$

with  $\xi(\phi) := \hat{\xi}_n + n^{-1/2} \phi$ . We see that  $h$  has to scale as  $O(n^{-1})$  in order for the noise to be of  $O(1)$  in the  $\phi$ -coordinate. Therefore we let  $h := c_1/n$  for some  $c_1 > 0$ .

The error of using the SGLD algorithm with a fixed step-size  $h_i = h$  is analysed in [Vollmer, Zygalakis and Teh \(2015\)](#). To first order the error is governed by the relative sizes of the variance of the estimator of the drift and the variance of the driving noise. Furthermore, it is possible to correct for this error providing the latter variance is greater than the former.

First we calculate the variance of the estimator of the drift. Define  $\sigma > 0$  by

$$\text{Var}(\nabla_{\xi} f(x^J \mid \xi)) = \sigma^2,$$

where the variance is with respect to the randomness induced by  $J$ , drawn uniformly among  $\{1, \dots, n\}$ . Then by the expression for the variance for sampling without replacement ([Rice, 2006](#), Section 7.3.1), and using  $\sqrt{n} \nabla_{\phi} \log \pi = \nabla_{\xi} \log \pi$ ,

$$\text{Var}\left(h \sqrt{n} \widehat{\nabla_{\xi} \log \pi}\right) = h^2 n^3 \text{Var}\left(\frac{1}{m} \sum_{i=1}^m \nabla_{\xi} (f(x^{J_i} \mid \xi))\right) = \frac{c_1^2 n \sigma^2}{m} \left(\frac{n-m}{n-1}\right).$$

This is  $O(n/m)$ . By comparison the variance of the driving noise is  $O(1)$ . If we want the former to be less than the latter we will need  $m$  to be  $O(n)$ . That is we will need to sub-sample a fixed proportion of the data at each iteration. Thus the advantage of SGLD over a method that does not use sub-sampling can at best be by a constant factor, and SGLD cannot be super-efficient. The only potential to develop SGLD to be super-efficient would be to substantially reduce the variance of the estimator of the drift, for example by using the control variate idea we use within ZZ-CV (see also [Huggins and Zou, 2016](#)).

#### 6.4. Bayesian inference on the mean of a Gaussian distribution

Consider the well known toy problem in Bayesian statistics of estimating the mean of a Gaussian distribution. This problem has the advantage that it allows for an analytical solution which can be compared with the numerical solutions obtained by Zig-Zag Sampling and other methods.

We assume that conditional on a parameter  $\xi \in \mathbb{R}^d$ , independent observations  $(x^j)_{j=1}^n$  have distribution  $N(\xi, \sigma^2)$ . We put a prior distribution  $\pi_0 \sim N(0, \rho^2)$  on  $\xi$ . This leads to a negative log density

$$\Psi(\xi) = \frac{\|\xi\|^2}{2\rho^2} + \frac{1}{2\sigma^2} \sum_{j=1}^n \|\xi - x^j\|^2, \quad \xi \in \mathbb{R}^d.$$

We compute

$$\nabla \Psi(\xi) = \left( \frac{1}{\rho^2} + \frac{n}{\sigma^2} \right) \xi - \frac{1}{\sigma^2} \sum_{j=1}^n x^j, \quad \xi \in \mathbb{R}^d$$

and

$$H_\Psi(\xi) = \left( \frac{1}{\rho^2} + \frac{n}{\sigma^2} \right) I, \quad \xi \in \mathbb{R}^d.$$

For any trajectory  $\xi(t) = \xi + \theta t$ , we have

$$\lambda_i(\xi(t), \theta) = \max(0, \theta_i \partial_i \Psi(\xi + \theta t)) = \max(0, a_i + b_i t) =: M_i(t),$$

with

$$a_i = \theta_i \frac{\xi_i}{\rho^2} + \frac{\theta_i}{\sigma^2} \sum_{j=1}^n [\xi_i - x_i^j] \quad \text{and} \quad b_i = \frac{1}{\rho^2} + \frac{n}{\sigma^2}, \quad i = 1, \dots, d.$$

We see that in this case we can construct computational bounds  $(M_i(t))$  which are exact, so that all proposed switching times will be accepted. The corresponding algorithm will be simply denoted by ZZ.

Since there is no global bound on the switching rate there is no straightforward way to implement the naive sub-sampling method of Section 4.2. However, because the Hessian of  $\Psi$  is constant it is possible to apply the sub-sampling method with control variates of Section 4.3. In fact, because the data has an additive effect on the gradient of  $\Psi$ , we have for arbitrary  $\xi^*$  that

$$E_i^j(\xi) = \partial_i \Psi(\xi^*) + \partial_i \Psi^j(\xi) - \partial_i \Psi^j(\xi^*) = \partial_i \Psi(\xi), \quad \xi \in \mathbb{R}^d,$$

we see that the sub-sampling switching rates are exactly equal to the canonical switching rates, and the two continuous time stochastic processes coincide, once we note that we can pre-compute  $\sum_{j=1}^n x^j$  in the expression for  $\nabla \Psi(\xi)$ .

However the computational bounds of the two algorithms are not equal, and it will be of interest to see how Zig-Zag with control variates behaves if  $\xi^*$  is not chosen to be exactly equal to the mode. The mode of  $\pi$  is given by

$$\xi^{\text{MAP}} = \frac{\frac{1}{n} \sum_{j=1}^n x^j}{1 + \frac{\sigma^2}{n\rho^2}},$$

and the one-off cost of computing this quantity is  $O(n)$ . Alternatively we can choose to use a sub-sampling of  $(x^j)$  to obtain a value  $\xi^*$  close to the posterior mode. As we require  $\xi^*$  to be within  $O(n^{-1/2})$  of the mode, the size of such a sample should be at least proportional to  $n$ . To be specific, we will consider the case where  $\xi^*$  is determined randomly by

$$\xi^* = \frac{\frac{1}{m} \sum_{j=1}^m x^{J_j}}{1 + \frac{\sigma^2}{n\rho^2}},$$

where  $m = \lceil cn \rceil$  for some constant  $c \in (0, 1]$ , and  $(J^i)_{i=1}^m$  are drawn randomly without replacement from  $\{1, \dots, m\}$ . The corresponding Zig-Zag algorithms are denoted by ZZ-soCV (sub-optimal Control Variates, for  $\xi^*$  an approximation to the mode), and ZZ-CV (for  $\xi^* = \xi^{\text{MAP}}$ ).

The constants  $(C_i)$  determining the computational bounds (as described in Section 4.3) are given by  $C_i = \frac{1}{\rho^2} + \frac{n}{\sigma^2}$  for  $i = 1, \dots, d$ , regardless of the choice of  $p \in [1, \infty]$ . Choosing  $p = \infty$  will give optimal scaling of  $a_i$  and  $b_i$  with respect to dimension in the computational bound  $M_i(t) = \max(0, a_i + b_i t)$ .

#### 6.4.1. Numerical comparison between Zig-Zag and SGLD for a Gaussian target distribution

In our first numerical experiment, we compare the mean square error (MSE) for several algorithms, namely basic Zig-Zag (ZZ), Zig-Zag with Control Variates (ZZ-CV), Zig-Zag with “sub-optimal” Control Variates (ZZ-soCV), and Stochastic Gradient Langevin Dynamics (SGLD). Here basic Zig-Zag refers to Zig-Zag, where we pretend that every iteration requires the evaluation of  $n$  observations (whereas in practice, we can pre-compute  $\xi^{\text{MAP}}$ ). Parameter values are  $\mu = 1$ ,  $\xi_0 = 1$  (for the true value of the mean parameter),  $\sigma = 1$  (specifying the Gaussian posterior distribution) and  $c_1 = 1$ ,  $c_2 = 1/10$  (for the SGLD parameters). The value of  $\xi^*$  for ZZ-soCV is based on a sub-sample of size  $m = n/10$ .

The MSE for the second moment using SGLD does not decrease beyond a fixed value, indicating the presence of bias in SGLD. This bias does not appear in the different versions of Zig-Zag sampling, agreeing with the theoretical result that ergodic averages over Zig-Zag trajectories are consistent. Furthermore we see a significant relative increase in efficiency for ZZ-(so)CV over basic ZZ when the number of observations is increased, agreeing with the scaling results of Section 5. In this experiment, the difference in MSE between ZZ-soCV and ZZ-CV is of the same order of magnitude as the relative size of the sub-sample of the data used in computing  $\xi^*$ , i.e. a factor 10.

#### 6.5. Logistic regression

Consider a binary data set  $y^j \in \{0, 1\}$ ,  $j = 1, \dots, n$  given  $d$ -dimensional covariates  $x^j \in \mathbb{R}^d$ ,  $j = 1, \dots, n$  (with  $x_1^j = 1$  for all  $j$ ) assumed to come from the

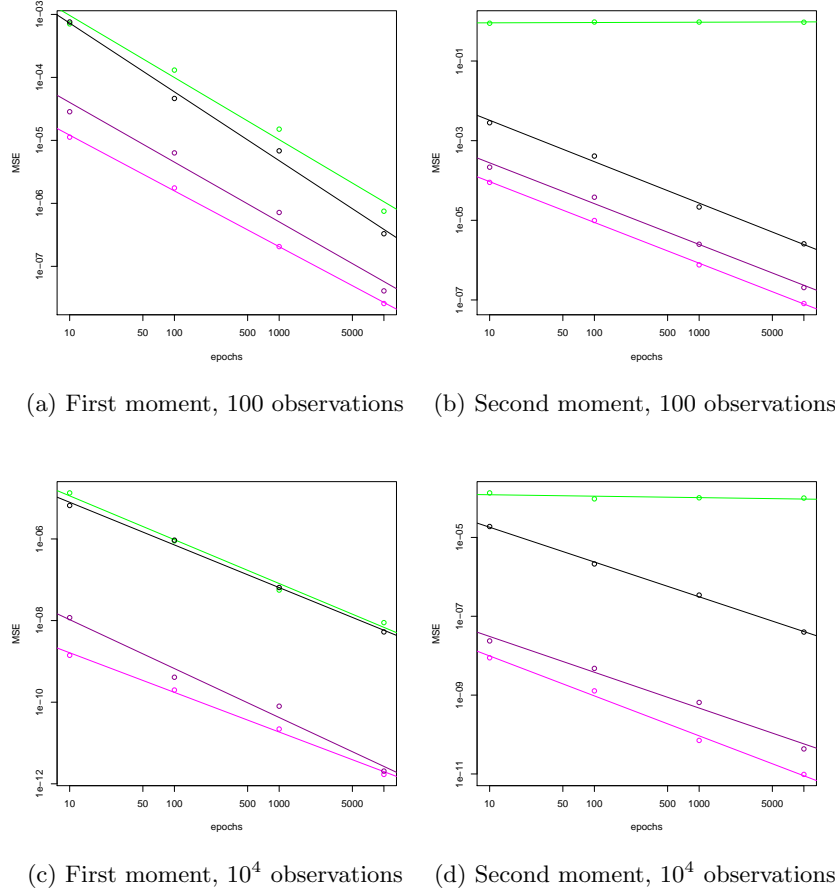


Figure 3: Log-log plots of the experimentally observed mean square error (MSE) in the first and second moment as a function of the number of epochs, based on  $n = 100$  or  $n = 10,000$  observations, for a one-dimensional Gaussian posterior distribution (Section 6.4). Displayed are SGLD (green), ZZ-CV (magenta), ZZ-soCV (dark magenta), ZZ (black). The displayed dots represent averages over experiments based on randomly generated data from the true posterior distribution.

logistic regression model,

$$\mathbb{P}(y = 1 \mid x, \xi) = \frac{1}{1 + \exp(-\sum_{i=1}^d \xi_i x_i)},$$

with parameter  $\xi \in \mathbb{R}^d$ . For any given prior probability distribution  $\pi_0$  on the value of  $\xi$ , we obtain for the posterior density function

$$\pi(\xi) = \pi_0(\xi) \prod_{j=1}^n \frac{\exp\left(y^j \sum_{i=1}^d x_i^j \xi_i\right)}{1 + \exp\left(\sum_{i=1}^d x_i^j \xi_i\right)}, \quad \xi \in \mathbb{R}^d.$$

For simplicity we assume a flat prior on  $\xi \in \mathbb{R}^d$ , i.e.  $\pi_0$  is constant. The corresponding negative log density function is now given by

$$\Psi(\xi) = \sum_{j=1}^n \left\{ \log \left( 1 + \exp \left( \sum_{i=1}^d x_i^j \xi_i \right) \right) - y^j \sum_{i=1}^d x_i^j \xi_i \right\}, \quad \xi \in \mathbb{R}^d.$$

For the  $k$ -th derivative we find

$$\partial_k \Psi(\xi) = \sum_{j=1}^n \left\{ \frac{x_k^j \exp\left(\sum_{i=1}^d x_i^j \xi_i\right)}{1 + \exp\left(\sum_{i=1}^d x_i^j \xi_i\right)} - y^j x_k^j \right\}, \quad \xi \in \mathbb{R}^d.$$

In order to prepare for sub-sampling, we can write  $\Psi = \frac{1}{n} \sum_{j=1}^n \Psi^j$ , with

$$\Psi^j(\xi) = n \log \left( 1 + \exp \left( \sum_{i=1}^d x_i^j \xi_i \right) \right) - n y^j \sum_{i=1}^d x_i^j \xi_i, \quad \xi \in \mathbb{R}^d, \quad j = 1, \dots, n.$$

We compute for  $j = 1, \dots, n$ ,

$$\partial_k \Psi^j(\xi) = \frac{n x_k^j \exp\left(\sum_{i=1}^d x_i^j \xi_i\right)}{1 + \exp\left(\sum_{i=1}^d x_i^j \xi_i\right)} - n y^j x_k^j, \quad \xi \in \mathbb{R}^d, \quad k = 1, \dots, d, \quad (16)$$

and

$$\partial_k \partial_l \Psi^j(\xi) = \frac{n x_k^j x_l^j \exp\left(\sum_{i=1}^d x_i^j \xi_i\right)}{\left(1 + \exp\left(\sum_{i=1}^d x_i^j \xi_i\right)\right)^2}, \quad \xi \in \mathbb{R}^d, \quad k, l = 1, \dots, d.$$

Using the estimate  $0 < \exp(a)/(1 + \exp(a)) < 1$ , we find that the global bound (13) holds with

$$c_i := n \max_{j=1, \dots, n} |x_i^j|,$$



so that we can use the sub-sampling method with a global bound on the switching rate, discussed in Section 4.2. Furthermore, using the bound  $\exp(a)/(1 + \exp(a))^2 \leq 1/4$ , we have

$$H_\Psi(\xi) \preceq Q := \frac{1}{4} \sum_{j=1}^n x^j (x^j)^\top,$$

so that we can use the Zig-Zag algorithm for dominated Hessian (without sub-sampling), discussed in Section 3.3. Finally, using analogous estimates, we find that

$$|\partial_k \partial_l \Psi^j(\xi)| \leq n |x_k^j x_l^j|/4, \quad \xi \in \mathbb{R}^d, \quad k, l = 1, \dots, d,$$

from which it follows that (14) is satisfied with

$$C_i := n \max_{j=1, \dots, n} \frac{1}{4} |x_i^j| \|x^j\|_2, \quad i = 1, \dots, d, \quad (17)$$

enabling the use of the sub-sampling method with control variates, discussed in Section 4.3.

*Remark 6.1.* If  $x^j$  are drawn independently from any (sub-)Gaussian distribution, taking the maximum in (17) results in  $C_i = O(n \log n)$ , using e.g. (Handel, 2014, Lemma 5.1). If on the other hand all  $x^j$  are taken (not necessary independently) from a bounded set, then trivially  $C_i = O(n)$ .

#### 6.5.1. ESS per epoch for logistic regression

In this numerical experiment we compare how the Effective Sample Size per epoch (ESSpE) grows with the number of observations  $n$  for several Zig-Zag algorithms. Recall from the discussion in Section 6.3 that for any MCMC which does not use sub-sampling, the ESSpE should be equal to a constant smaller than one, and if, it existed, an algorithm able to generate an independent sample by processing all data would have an ESSpE exactly equal to one. The results of this experiment are shown in Figure 4. In both the plots of ESS per epoch (see (a) and (c)), the best linear fit for ZZ-CV has slope approximately 0.95, which is in close agreement with the scaling analysis of Section 5. The other algorithms have roughly a horizontal slope, corresponding to a linear scaling with the size of the data. As a result, ZZ-CV is the only algorithm for which the ESS per CPU second is approximately constant as a function of the size of the data (see (b) and (d)). Hence we see ZZ-CV obtains an ESSpE which is roughly linearly increasing with the number of observations  $n$ . The other versions of the Zig-Zag algorithms have a ESSpE which is approximately constant with respect to  $n$ . These statements apply regardless of the dimensionality of the problem.

## Appendix A: Proofs for results related to ergodicity

**Lemma A.1.** Suppose  $\pi^k$  is given as the product of measures,  $\pi^k := \pi_1^k \otimes \dots \otimes \pi_d^k$ , with  $\pi_i^k$  probability measures on Borel spaces  $E_i$  for  $k \in \mathbb{N}$  and fixed

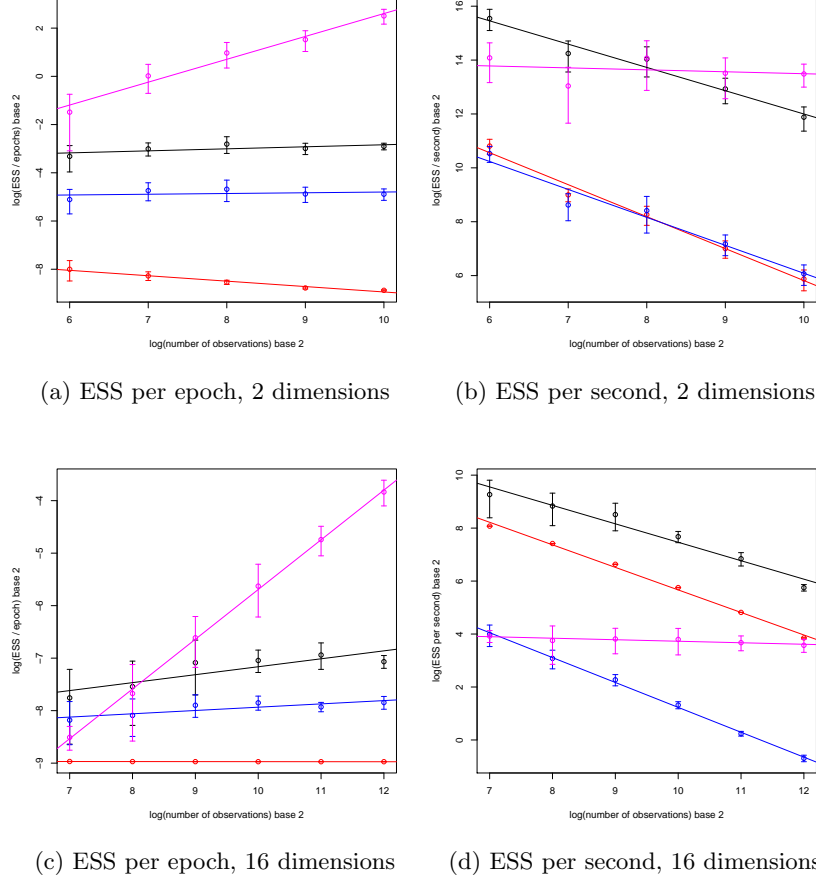


Figure 4: Log-log plots of the experimentally observed dependence of ESS per epoch (ESSpE) and ESS per second (ESSpS) as a function of the number of observations  $n$  in the case of Bayesian logistic regression. Experimental results for logistic regression, see Section 6.5. In these experiments various versions of the Zig-Zag algorithm are run for  $10^5$  epochs on a 2-dimensional and a 16-dimensional logistic regression problem with randomly generated data based on true parameter values  $\xi = (2, 1)$  and  $\xi = (1, \dots, 1)$ , respectively. Plotted are mean and standard deviation over 10 experiments, along with the best linear fit. Displayed are Zig-Zag with global bound (red), Zig-Zag with Lipschitz bound (black), Zig-Zag with sub-sampling using global bound (blue) and Zig-Zag with Control Variates (ZZ-CV, magenta). The experiments were carried out on a 2013 laptop computer.

$d \in \mathbb{N}$ ,  $d \geq 2$ . Suppose for every  $i = 1, \dots, d$  there exists a measure  $\pi_i$  such that  $\lim_{k \rightarrow \infty} \|\pi_i^k - \pi_i\|_{\text{TV}} = 0$ . Then

$$\lim_{k \rightarrow \infty} \|\pi^k - \bigotimes_{i=1}^d \pi_i\|_{\text{TV}} = 0.$$

For  $f \in C_b(E)$ , with  $E$  a topological space, let  $\|f\|$  denote the supremum norm of  $f$ .

*Proof.* It suffices to prove the result for  $d = 2$ . Write  $E = E_1 \times E_2$ . For  $f \in C_b(E)$ ,  $\|f\| \leq 1$ , define  $h_f^k(x_1) := \int_{E_2} f(x_1, y_2) d\pi_2^k(y_2)$ . Note that for any such  $f$  and  $k \in \mathbb{N}$ ,  $h_f^k \in C_b(E_1)$  with  $\|h_f^k\| \leq 1$ . Therefore, as  $k \rightarrow \infty$ ,

$$\begin{aligned} & \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} |(\pi_1^k \otimes \pi_2^k)(f) - \pi_1(h_f^k)| \\ &= \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} \left| \int_{E_1} \int_{E_2} f(y_1, y_2) d\pi_1^k(y_1) d\pi_2^k(y_2) - \pi_1(h_f^k) \right| \\ &= \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} \left| \int_{E_1} h_f^k d\pi_1^k - \pi_1(h_f^k) \right| \leq \sup_{\substack{h \in C_b(E) \\ \|h\| \leq 1}} \left| \int_{E_1} h d\pi_1^k - \pi_1(h) \right| \rightarrow 0. \end{aligned} \tag{18}$$

Also, for any  $y_1 \in E_1$ ,  $f(y_1, \cdot) \in C_b(E_2)$  with supremum norm less than or equal to one. Therefore, for all  $y_1 \in E_1$ , as  $k \rightarrow \infty$ ,

$$\begin{aligned} H^k(y_1) &:= \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} \left| \int_{E_2} f(y_1, y_2) d\pi_2^k(y_2) - \int_{E_2} f(y_1, y_2) \pi(dy_2) \right| \\ &\leq \sup_{\substack{g \in C_b(E_2) \\ \|g\| \leq 1}} \left| \int_{E_2} g d\pi_2^k - \int_{E_2} g d\pi_2 \right| \rightarrow 0. \end{aligned}$$

Hence by bounded convergence, as  $k \rightarrow \infty$ ,

$$\begin{aligned} & \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} |\pi_1(h_f^k) - (\pi_1 \otimes \pi_2)(f)| \\ &= \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} \left| \int_{E_1} \left\{ \int_{E_2} f(y_1, y_2) d\pi_2^k(y_2) - \int_{E_2} f(y_1, y_2) d\pi_2(y_2) \right\} d\pi_1(y_1) \right| \\ &\leq \int_{E_1} \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} \left| \int_{E_2} f(y_1, y_2) d\pi_2^k(y_2) - \int_{E_2} f(y_1, y_2) d\pi_2(y_2) \right| d\pi_1(y_1) \\ &= \int_{E_1} H^k d\pi_1 \rightarrow 0. \end{aligned} \tag{19}$$

Combining (18) and (19) gives, for any  $(x_1, x_2) \in E$ ,

$$\begin{aligned} \|\pi^k - \pi_1 \otimes \pi_2\|_{\text{TV}} &= \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} |\pi^k(f) - (\pi_1 \otimes \pi_2)(f)| \\ &\leq \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} |(\pi_1^k \otimes \pi_2^k)(f) - \pi_1(h_f^k)| + \sup_{\substack{f \in C_b(E) \\ \|f\| \leq 1}} |\pi_1(h_f^k) - (\pi_1 \otimes \pi_2)(f)| \rightarrow 0. \end{aligned}$$

□

A discrete time Markov chain in  $E$  with transition kernel  $P$  is called  $\varphi$ -irreducible if there exists a non-trivial Borel measure  $\varphi$  on  $E$  such that, whenever  $\varphi(A) > 0$  for  $A \in \mathcal{B}(E)$  and  $x \in E$ , there exists a  $k \in \mathbb{N}$  such that  $P^k(x, A) > 0$ .

**Lemma A.2.** *Suppose the Markov chain on  $E$  with transition kernel  $P(x, dy)$  is mixing with respect to its unique invariant probability distribution  $\pi$ . Then the transition kernel  $P$  is  $\pi$ -irreducible.*

*Proof.* Let  $A \in \mathcal{E}$  such that  $\pi(A) > 0$ . Since the Markov chain is mixing, there exists a  $k$  such that  $|P^k(x, A) - \pi(A)| < \pi(A)/2$ , so that  $P^k(x, A) > 0$ . □

**Proof of Theorem 2.11.** Let  $(N_1(t), \dots, N_d(t))$  denote  $d$  independent Poisson processes, each with constant rate  $\gamma > 0$  defined on a filtered probability space  $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{Q})$ . Given  $(\xi, \theta) \in E$ , define a stochastic processes  $\Theta_i(t; \xi, \theta) := (-1)^{N_i(t)} \theta_i$  for  $i = 1, \dots, d$  and let  $\Xi_i(t; \xi, \theta) := x_i + \int_0^t \Theta_i(s; \theta) ds$ . Then under  $\mathbb{Q}$ ,  $(\Xi(\cdot; \xi, \theta), \Theta(\cdot; \xi, \theta))$  corresponds to a Zig-Zag process started in  $(\xi, \theta)$  with constant switching rate  $\gamma$ . Denote the transition kernel for this process by  $Q^t((\xi, \theta), \cdot)$ .

Write  $\lambda(s; \xi, \theta) := \lambda(\Xi(s; \xi, \theta), \Theta(s; \xi, \theta))$ . For  $(\xi, \theta) \in E$  define a stochastic process  $Z(t; \xi, \theta)$  on  $(\Omega, \mathcal{F}, (\mathcal{F}_t))$  by

$$\begin{aligned} Z(t; \xi, \theta) &= \exp \left( \sum_{i=1}^d \int_0^t \log \left( \frac{\lambda_i(s; \xi, \theta)}{\gamma} \right) dN_i(s) - \sum_{i=1}^d \int_0^t \{\lambda_i(s; \xi, \theta) - \gamma\} ds \right), \end{aligned}$$

Since  $\lambda_i(s; \xi, \theta) > 0$  for all  $i = 1, \dots, d$ ,  $s \geq 0$ , and  $(\xi, \theta) \in E$ , it follows that  $\lambda_i(\Xi(s; \xi, \theta), \Theta(s; \xi, \theta))$  is bounded away from 0 for all  $i = 1, \dots, d$  and  $0 \leq s \leq t$ . Using this local boundedness property the processes  $(Z(\cdot; \xi, \theta))$  are a.s. positive martingales. For fixed  $(\xi, \theta)$ , the probability measure  $\mathbb{P}_{\xi, \theta}$  on  $\mathcal{F}_t$ , defined by the Radon-Nikodym derivative

$$\left. \frac{d\mathbb{P}_{\xi, \theta}}{d\mathbb{Q}} \right|_{\mathcal{F}_t} = Z(t; \xi, \theta)$$

is such that under  $\mathbb{P}_{\xi, \theta}$ , the processes  $N_i(s)$  have time inhomogeneous rate  $\lambda_i(\Xi(s; \xi, \theta), \Theta(s; \xi, \theta))$ . Let  $P^t((\xi, \theta), \cdot)$  denote the probability distribution of  $(\Xi(t), \Theta(t))$  under  $\mathbb{P}$ , and similarly  $Q^t$  for the distribution under  $\mathbb{Q}$  for  $t \geq 0$ . Then

$$P^t((\xi, \theta), A) = \mathbb{E}^{\mathbb{Q}} [Z(t; \xi, \theta) \mathbb{1}_A(\Xi(t; \xi, \theta), \Theta(t; \xi, \theta))],$$

whence for all  $(\xi, \theta)$  and  $t \geq 0$ ,  $P^t((\xi, \theta), \cdot)$  and  $Q^t((\xi, \theta), \cdot)$  are equivalent.

Now take  $\tilde{\lambda}$  to be equal to the switching rates for a standard normal target distribution with excessive switching rate  $\gamma$ , i.e.

$$\tilde{\lambda}_i(\xi, \theta) = (\theta_i x_i)^+ + \gamma,$$

and repeat the above construction to obtain transition probabilities  $\tilde{P}^t((\xi, \theta), \cdot)$ . It follows that the transition probabilities  $P^t$  and  $\tilde{P}^t$  are equivalent for all  $t \geq 0$  and  $(\xi, \theta) \in E$ . From Proposition 2.9 and Example 2.10 it follows that the time discretization of the Zig-Zag process with transition kernels  $(\tilde{P}^{\delta k})$  is mixing. By Lemma A.2, it follows that the transition kernels  $(\tilde{P}^{\delta k})$  correspond to a  $\varphi$ -irreducible process. By the equivalence of the transition kernels  $P^t$  and  $\tilde{P}^t$ , this property carries over to the Zig-Zag process with switching rates  $\lambda$ . It follows that there can be at most a single unique invariant distribution for the time discretization of the Zig-Zag process, and this property carries over to the continuous time Zig-Zag process.  $\square$

## Bibliography

- ANDRIEU, C. and ROBERTS, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics* **37** 697–725.
- BIERKENS, J. (2015). Non-reversible Metropolis-Hastings. *Statistics and Computing* **25** 1–16.
- BIERKENS, J. and ROBERTS, G. (2016). A piecewise deterministic scaling limit of Lifted Metropolis-Hastings in the Curie-Weiss model. *arXiv preprint arXiv:1509.00302*. To appear in *Annals of Applied Probability*.
- BOUCHARD-CÔTÉ, A., VOLLMER, S. J. and DOUCET, A. (2015). The Bouncy Particle Sampler: A Non-Reversible Rejection-Free Markov Chain Monte Carlo Method. *arXiv:1510.02451*.
- CHEN, T.-L. and HWANG, C.-R. (2013). Accelerating reversible Markov chains. *Statistics & Probability Letters* **83** 1956–1962.
- DAVIS, M. H. A. (1984). Piecewise-Deterministic Markov Processes: A General Class of Non-Diffusion Stochastic Models. *Journal of the Royal Statistical Society. Series B (Methodological)* **46** 353–388.
- DIACONIS, P., HOLMES, S. and NEAL, R. (2000). Analysis of a nonreversible Markov chain sampler. *Annals of Applied Probability* **10** 726–752.
- DUANE, S., KENNEDY, A. D., PENDLETON, B. J. and ROWETH, D. (1987). Hybrid Monte Carlo. *Physics Letters B* **195** 216–222.
- DUNCAN, A. B., LELIÈVRE, T. and PAVLIOTIS, G. A. (2016). Variance Reduction using Nonreversible Langevin Samplers. *Journal of Statistical Physics* **163** 457–491.
- ETHIER, S. N. and KURTZ, T. G. (2005). *Markov Processes: Characterization and Convergence (Wiley Series in Probability and Statistics)*. Wiley-Interscience.

- FONTBONA, J., GUÉRIN, H. and MALRIEU, F. (2012). Quantitative estimates for the long-time behavior of an ergodic variant of the telegraph process. *Advances in Applied Probability* **44** 977–994.
- FONTBONA, J., GUÉRIN, H. and MALRIEU, F. (2016). Long time behavior of Telegraph Processes under convex potentials. *Stochastic Processes and their Applications*. in press.
- GOLDSTEIN, S. (1951). On diffusion by discontinuous movements, and on the telegraph equation. *Quarterly Journal of Mechanics and Applied Mathematics* **4** 129–156.
- HANDEL, R. V. (2014). Probability in High Dimension. <http://www.princeton.edu/~rvan/ORF570.pdf>.
- HASTINGS, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57** 97–109.
- HUGGINS, J. H. and ZOU, J. (2016). Quantifying the accuracy of approximate diffusions and Markov chains. *arXiv:1605.06420*.
- HWANG, C., HWANG-MA, S. and SHEU, S. (1993). Accelerating Gaussian diffusions. *The Annals of Applied Probability* **3** 897–913.
- JACOB, P. E. and THIERY, A. H. (2015). On nonnegative unbiased estimators. *The Annals of Statistics* **43** 769–784.
- JOHNSON, R. A. (1970). Asymptotic Expansions Associated with Posterior Distributions. *Annals of Mathematical Statistics* **41** 851–864.
- KAC, M. (1974). A stochastic model related to the telegrapher’s equation. *Rocky Mountain J. Math.* **4** 497–509.
- LELIEVRE, T., NIER, F. and PAVLIOTIS, G. A. (2013). Optimal Non-reversible Linear Drift for the Convergence to Equilibrium of a Diffusion. *Journal of Statistical Physics* **152** 237–274.
- LEWIS, P. A. W. and SHEDLER, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval Res. Logist. Quart.* **26** 403–413.
- MA, Y.-A., CHEN, T. and FOX, E. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems* 2917–2925.
- METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. and TELLER, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* **21** 1087.
- MINSKER, S., SRIVASTAVA, S., LIN, L. and DUNSON, D. B. (2014). Robust and Scalable Bayes via a Median of Subset Posterior Measures. *Kinetic and Related Models* **2** 341–360.
- MONMARCHÉ, P. (2014). Hypocoercive relaxation to equilibrium for some kinetic models via a third order differential inequality. *arXiv:1306.4548*.
- NEAL, R. M. (1998). Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. In *Learning in graphical models* 205–228. Springer.
- NEAL, R. (2011). MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, **2** 113–162.
- NEISWANGER, W., WANG, C. and XING, E. (2013). Asymptotically Exact, Embarrassingly Parallel MCMC. *arXiv:1311.4780*.

- PETERS, E. A. J. F. and DE WITH, G. (2012). Rejection-free Monte Carlo sampling for general potentials. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* **85** 1–5.
- POLLOCK, M., FEARNHEAD, P., JOHANSEN, A. J. and ROBERTS, G. O. (2016). An Unbiased and Scalable Monte Carlo Method for Bayesian Inference for Big Data. *in preparation*. see <http://www.birs.ca/events/2015/5-day-workshops/15w5160/videos/watch/201506021533-Roberts.html>.
- QUIROZ, M., VILLANI, M. and KOHN, R. (2015). Speeding up MCMC by efficient data subsampling. *Riksbank Research Paper Series* **121**.
- REY-BELLET, L. and SPILIOPOULOS, K. (2015). Irreversible Langevin samplers and variance reduction: a large deviations approach. *Nonlinearity* **28** 2081–2103.
- RICE, J. (2006). *Mathematical statistics and data analysis*. Nelson Education.
- SCOTT, S. L., BLOCKER, A. W. and BONASSI, F. V. (2016). Bayes and Big Data: The Consensus Monte Carlo Algorithm. *International Journal of Management Science and Engineering Management* **11** 78–88.
- SRIVASTAVA, S., CEVHER, V., TRAN-DINH, Q. and DUNSON, D. B. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. In *AISTATS*.
- SUN, Y., GOMEZ, F. and SCHMIDHUBER, J. (2010). Improving the Asymptotic Performance of Markov Chain Monte-Carlo by Inserting Vortices. In *Advances in Neural Information Processing Systems 23* (J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta, eds.) 2235–2243.
- TEH, Y. W., THIERY, A. H. and VOLLMER, S. (2014). Consistency and fluctuations for stochastic gradient Langevin dynamics. *arXiv:1409.0578*.
- TURITSYN, K. S., CHERTKOV, M. and VUCELJA, M. (2011). Irreversible Monte Carlo algorithms for efficient sampling. *Physica D: Nonlinear Phenomena* **240** 410–414.
- VOLLMER, S. J., ZYGALAKIS, K. C. and TEH, Y. W. (2015). (Non-) asymptotic properties of Stochastic Gradient Langevin Dynamics. *arXiv:1501.00438*.
- WANG, X. and DUNSON, D. B. (2013). Parallelizing MCMC via Weierstrass Sampler. *arXiv:1312.4605*.
- WELLING, M. and TEH, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* 681–688.